Bachelor Thesis

Which classifier is fairer? Comparing the effects of distribution shifts on subgroup fairness

Author

Theresa Kriecherbauer

Supervisor

Prof. Dr. Christoph Kern
Department of Statistics

Department of Statistics

Ludwig-Maximilians-Universität München

Munich, 22th of December 2022

Abstract

The performance of machine learning models can vary greatly for different subgroups based on sensitive variables like age, nationality or gender. When applied in contexts such as medical diagnosis, these models can be regarded as unfair since misclassification may have a strongly negative impact on someone's life. In this thesis, we exploratively investigate the role that the choice of the machine learning model plays for subgroup fairness - which has so far been out of focus - and how this role changes when the distribution of subgroups shifts. This is done for four commonly used machine learning classifiers - logistic regression, random forests, support vector machines and XGBoost - on one specific tabular dataset, the UCI Adult dataset. We first summarise model properties, that could potentially influence subgroup performance. We then generate one unshifted and three shifted training samples by simulating sample selection bias. Next, we descriptively evaluate whether the classifiers differ regarding our newly introduced subgroup fairness notion of *Subgroup Performance Parity* which is defined for Accuracy and Matthews Correlation Coefficient as performance measures. We observe that both the role of the model choice and the distribution shifts have limited influence on overall subgroup fairness. Support vector machines appear to be more subgroup-fair in the unshifted case. However, this occurs only by a small margin and only for Matthews Correlation Coefficient.

Contents

List of Figures

List of Tables

1 Introduction

In numerous situations such as medical diagnosis, machine learning models have a higher performance in decision-making than humans (Antonelli et al., 2019). However, this performance can vary greatly for individuals depending on subgroup membership based on sensitive variables such as age, nationality or gender, as the following two examples show. In the field of psychology, so called happiness statements are more often misclassified for some demographic subgroups such as for non-parent males (Sweeney et al., 2021). In the area of face recognition, Buolamwini & Gebru (2018) have found that the results with commercial software vary notably for different race-gender combinations. These models are used to make decisions that could substantially influence the lives of individuals in some situations. Since not all subgroups experience the same quality in classification by the machine learning models, the two applications just presented can therefore be said to be unfair at subgroup level. While the influence of data has been widely discussed as a possible factor in such inequalities, the role of machine learning models has been rarely studied as another possible cause, as Hooker (2021) explains. This thesis will therefore exploratively investigate the role that the choice of the machine learning model plays for subgroup fairness for one specific tabular dataset - the UCI Adult dataset.

One factor that influences a classifier's performance - and therefore potentially also subgroup performance - is whether the model is deployed in a context similar to the one it was exposed to during training (Kull & Flach, 2014). In such a situation, distributions might differ which is referred to as dataset shift or distribution shift as in Koh et al. (2021). Distribution shifts usually negatively impact the classifier's performance. Therefore, we will also explore whether and how the subgroup fairness of different models changes, when the distribution of subgroups differs. To answer the two aspects of our research question, we study the performance of four common machine learning classifiers - logistic regression, random forests, support vector machines and XGBoost - on the subgroup level for different degrees of distribution shift. As the UCI Adult dataset represents a binary classification problem, the methods in this paper will be described for the binary type only.

This thesis is structured as follows. Section 2 provides an introduction to the field of fairness research in machine learning and defines the terms *subgroup fairness* and *distribution shift* as used in this thesis. Section 3 summarises related research. Section 4 outlines the methods needed to practically analyse our research scope described above. This includes details on the dataset, its pre-processing and the generation of the distribution shifts. In addition, it presents the four machine learning models as well as the procedure for their descriptive analysis with regard to subgroup fairness. In Section 5, the models are trained on differently distributed samples drawn from the dataset and are descriptively evaluated regarding their subgroup fairness. Section 6 discusses the main findings and limitations. Section 7 concludes.

2 Terminology

The introductory examples have shown the relevance of *subgroup-fair* machine learning also in the case of *distribution shifts*. We will now formalise these two concepts for our investigation.

2.1 Fairness in Machine Learning

Quantifying fairness has a long tradition, as Hutchinson & Mitchell (2018) point out. They explain that the introduction of the United States Civil Rights Act in 1964 sparked a debate about the fairness of assessment tests for hiring. This led to the first mathematical formalisations of fairness, which in many cases resemble today's proposals. Depending on the context, a variety of fairness definitions have since been developed in the area of machine learning that Mehrabi et al. (2019) divide into three groups: group fairness, individual fairness[1] and, more recently, subgroup fairness. Group fairness notions usually aim to equalise a given statistical measure among a ”small number of protected demographic groups” which can be done ”without making any assumptions about the underlying population” (Kearns et al., 2018, p. 1). However, focusing on a single attribute to obtain a fair classification can lead to discrimination regarding other attributes, as they illustrate. For this reason, concepts of individual fairness have been developed, which stipulate that similar individuals must be treated similarly (Kearns et al., 2018). But among other problems, Fleisher (2021) points out that the measurement of similarity between two individuals depends on the similarity metric chosen, which in turn is susceptible to human bias.

2.2 Subgroup Fairness

Based on these challenges with respect to the notions of group and individual fairness, Kearns et al. (2018) and Hebert-Johnson et al. (2018) independently introduced the idea of subgroup fairness which was developed with the intention of combining the above mentioned advantages of individual and group fairness notions as Kearns et al. (2018) explain. Kearns et al. (2018) suggest to more granularly divide the population into groups based on multiple sensitive variables, which they refer to as subgroups. On these subgroups they apply notions of group fairness such as equality of opportunity or statistical parity. They also propose algorithms to achieve these notions of fairness for a classifier and practically demonstrated the efficiency of these algorithms (Kearns et al., 2019). Hebert-Johnson et al. (2018) use a very similar formal setup but with a different group fairness notion - Calibration - as Kearns et al. (2018) explain. In the context of multiple subgroups they refer to this notion as Multicalibration. This work inspired some of the contributors to develop two further fairness definitions for subgroups. First, Kim, Ghorbani, & Zou (2018) presented an algorithm for obtaining Multiaccuracy, a concept already

[1] Caton & Haas (2020) provide a comprehensive overview of the most common group and individual fairness concepts and further classification approaches.

briefly mentioned by Hebert-Johnson et al. (2018) as a weaker notion compared to Multicalibration. In addition, Kim and his colleagues put forward a concept of subgroup fairness based on individual fairness notions, as opposed to ideas drawn from group fairness definitions in the earlier work on subgroup fairness. They propose a notion called Metric-Multifairness, which leads to a similar treatment of similar subpopulations (Kim, Reingold, & Rothblum, 2018). In contrast to other individual fairness notions, only some similarity metric values between pairs of individuals are specified in advance.

While other researchers acknowledge the underlying concept, they also point out the drawbacks of the initial idea of Kearns et al. (2018) and Hebert-Johnson et al. (2018). We have identified two important fairness notions related to population subgroups in our paper review, that have been developed in response. First, in 2019, Foulds et al. (2020) introduced the idea of intersectional fairness. This idea is based on research about discrimination in the humanities, which usually takes into account several dimensions of discrimination such as race, gender and sexual orientation. They define five criteria that can be derived from this principle and show that the proposals of Kearns et al. (2018) and Hebert-Johnson et al. (2018) do not meet them. Therefore, they propose a new notion - differential fairness - that satisfies the criteria of intersectional fairness. Second, N. L. Martinez et al. (2021) proposed to apply the group fairness notion of Minimax Pareto Fairness, inspired by the Rawlsian minimax idea and Pareto efficiency, that they had previously introduced on all possible subgroups of a certain predefined size (N. Martinez et al., 2020). The advantage of this approach is that sensitive attributes do not have to be known. Instead the worst-case risk across all possible subgroups is minimised. Their work is closely related to Hashimoto et al. (2018) and Balashankar et al. (2019).

2.3 Subgroup Performance Parity

The last two subsections have given a comprehensive overview of the understanding of fairness in machine learning and different ideas on how to define it on subgroup level. In this thesis, we consider equal performance on the subgroup level as a form of fairness which was motivated by the introductory examples. The above-mentioned subgroup fairness notion of Multiaccuracy probably is intuitively most closely related to our understanding of subgroup fairness. However, we would like to include an additional performance measure, the Matthews Correlation Coefficient as discussed in Section 4.4.1. It might be more appropriate in situations where the data is highly imbalanced (Fernández et al., 2018). Therefore, we introduce the notion of Subgroup Performance Parity and refer to this definition whenever we use the term of *subgroup fairness*. In our formal setup, we include notations from Kearns et al. (2018) and Kull & Flach (2014).

A given training sample can be notated as a matrix T where each column corresponds to a variable and each row represents an individual. This matrix can be divided into three submatrices based on the columns. The submatrix Z contains all the sensitive variables where a row z represents the vector of sensitive attributes of an individual. The submatrix X contains all the non-sensitive variables where a row x represents the vector of non-sensitive attributes of an individual. The submatrix Y contains the target variable where a row y represents the binary value of the target variable of an individual. We assume that the sample matrix T or equivalently notated as (X, Y, Z) was drawn from the true joint distribution of the population (X_t, Y_t, Z_t) where a random variable M, that represents the sampling procedure determines, whether an observation is drawn into the sample $(M = 1)$ or not $(M = 0)$.

Each individual is part of exactly one subgroup $i \in \{1, ..., I\}$, $I \in \mathbf{N}$ based only on its sensitive attributes z which can be expressed by the functions $g_i : SZ \to \{0, 1\}$, $g_i \in G$, where SZ denotes the set of all vectors z with $g_i(z) = 1$ indicating membership in subgroup i and G denoting a family of indicator functions.

We suggest two criteria that define Subgroup Performance Parity for a performance measure PM, which can be either the Accuracy or Matthews Correlation Coefficient as discussed in Section 4.4.1.

On the one hand, for a classifier D, the difference between its best and worst performing subgroup should not exceed a desirable value $\alpha \in \mathbb{R}_{\geq 0}$

$$||max_i(PM(D, i))| - |min_i(PM(D, i))|| < \alpha \tag{2.1}$$

where $PM(D, i)$ denotes the performance value for classifier D and subgroup i depending on the performance measure PM. We will not further discuss the choice of α, since the focus of this thesis is to compare classifiers among each other.

The second criterion should take into account by how much the performance between subgroups varies, since the difference in Equation 2.1 could be influenced by an outlier. Therefore, we propose computing the variance of all subgroup performance values as a an additional measure, where the value of $\epsilon \in \mathbb{R}_{\geq 0}$ presents the desired value:

$$\sqrt{V((PM(D, 1), \, ..., \, PM(D, I))^T)} < \epsilon \tag{2.2}$$

A classifier D satisfies Subgroup Performance Parity $SPP(\alpha, \epsilon)$ when for a fixed α and ϵ the two criteria are fulfilled. This subgroup fairness notion can also be used for comparison between multiple classifiers. One of them is superior over the others in terms of subgroup fairness, when its values for both criterion 1 and 2 are smaller than those of the other classifiers. This approach will be used for the analysis in Section 5.

2.4 Distribution Shift

Distribution shifts may have multiple causes as Storkey (2009) explains. One of them is the sample selection bias, that we simulate in a slightly adapted way in our analysis. It is present when "the sample of instances that we observe in training is biased compared to testing" (Kull & Flach, 2014, p. 6). They outline different types of sampling bias that can be distinguished according to the underlying reason. We base our definition of sample selection bias on these ideas but extend it by differentiating between the sensitive and non-sensitive variables of an individual. In our case, we assume that a membership in a certain subgroup i influences how likely it is for individuals to be drawn into the sample. Therefore, the causal relationship can be illustrated as in the following figure.

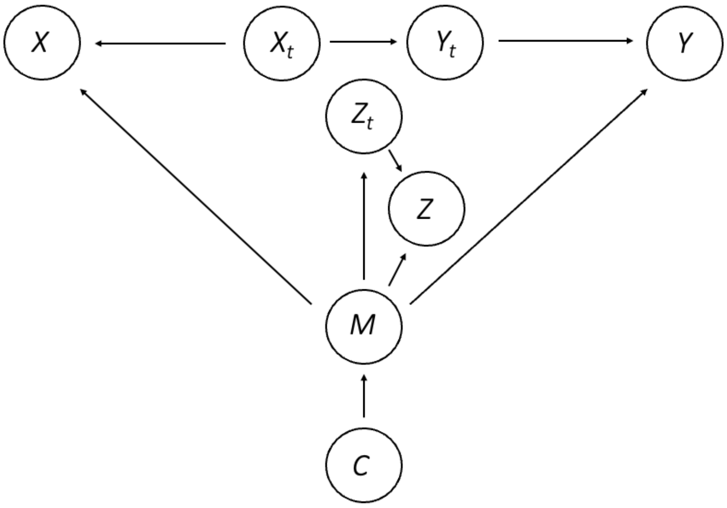

Figure 1: Illustration of the causal relationship between the true distribution and the training sample, based on Kull & Flach (2014).

We use the same notation as in the previous section and assume that (X_t, Z_t, Y_t) represents the true joint distribution of the sensitive and non-sensitive variables of the population as well as the binary target variable. As in any sampling procedure, a random variable M determines whether the observation is drawn into the sample ($M = 1$) or not ($M = 0$). However, the sampling probabilities depend on the membership of individuals in a subgroup $i \in \{1, ..., I\}$ and therefore the sensitive variables Z_t. These sampling probabilities vary depending on the context C. Therefore, we obtain a training sample distribution (X, Z, Y) that is different from the true distribution. For an overview of other types of distribution shifts, see Morreno-Torres et al. (2012) and Storkey (2009).

3 Related Work

As already pointed out, on the one hand, we want to explore the relationship between the model choice and subgroup fairness. On the other hand, we aim to study whether and how this relationship changes under distribution shift. In the next two sections, we summarise previous research undertaken in these two areas.

3.1 Relationship between Model Choice and Subgroup Fairness

The role of the model choice for subgroup fairness has rarely been studied systematically[2]. Instead, researchers have focused on the development of new methods to achieve subgroup-fair models, however, without examining the problems and properties of existing models in more depth. These fairness-enhancing algorithms can be divided into three groups as Friedler et al. (2018) point out: pre-processing methods, algorithm modifications that are applied during training and post-processing methods. When it comes to subgroup-fair methods, the methods we came across mostly fall into the two latter categories. These include algorithms that researchers conceived to achieve fairness with respect to their subgroup fairness definitions that we have reviewed in Section 2.2 such as Kearns et al. (2018), Hebert-Johnson et al. (2018) and N. L. Martinez et al. (2021). Besides, other approaches have been proposed. For instance, Lahoti et al. (2020) explain how adversarially reweighted learning can be used to train subgroup-fair models when the sensitive variables are unknown. Creager et al. (2019) suggest to use concepts from disentangled representation learning to train a subgroup-fair model that can be adapted based on the test dataset. Two papers which examine a research question similar to the one in this thesis are summarised in the next two paragraphs.

Bono et al. (2021) compare a traditional logistic model with two more sophisticated machine learning models on a credit scoring task. They have chosen an *extremely random forest* model and XGBoost. These models are trained on data from 800,000 UK borrowers. Subgroups are formed based on the sensitive variables gender, race and vulnerability that are not included in the model. The models are then evaluated per subgroup using the AUC and Average Accuracy as performance measures. One of their results is particularly interesting in terms of subgroup fairness. They find that the two machine learning models neither close nor worsen the performance gap between subgroups compared to the basic logistic model.

Gardner et al. (2022) see similarities between the underlying ideas of fairness and robustness. Both aim at reducing the subgroup performance variation and at maximising the performance of the worst subgroup, as they point out. Therefore, they compare four different types of model classes: fairness-enhancing models such as in-processing methods, robust models such as distributionally robust optimization, tree-based models such as random forest or XGBoost and

[2] The literature review method for this survey of related work is documented in A.1.

baseline models such as support vector machines. They find that tree-based models in almost all studied cases perform better with respect to the two above criteria than models specifically optimized for fairness or robustness and the baseline models. Additionally, tree-based models are less sensitive to hyperparameter changes than other models. The underlying idea of this work and the measurement of subgroup fairness has some similarities with our approach, however has been developed independently.

If we remove the restriction that the papers considered here deal only with subgroups, we can identify other papers that examine how certain dataset properties influence the performance of classifiers. Six dataset properties have been identified that classifiers react to differently.

1. **Dataset Complexity** Fernández-Delgado et al. (2014) investigate dataset complexity - which they admit is difficult to objectify - by evaluating 179 classifiers across 121 datasets. They assume that datasets that are classified less accurately by all evaluated classifiers must generally be more difficult to analyse. Therefore, they calculate weighted accuracies per classifier with more difficult datasets carrying higher weights. The maximum accuracy achieved per dataset serves as a proxy for the dataset's complexity. They find that the parallel implementation of random forests performs best on complex datasets, followed closely by other specifications of random forests and support vector machines with Gaussian kernels implemented in C and Weka. However, as Wainberg et al. (2016) note, this difference is not statistically significant. Furthermore, they point out that the results may be biased because the classifiers were not evaluated on a separate validation dataset. XGBoost was not included in their work.

2. **Number of Observations** Zheng & Jin (2020) and Fernández-Delgado et al. (2014) also study the influence of the training data size. Zheng & Jin (2020) who compare ensemble classifiers to traditional models on nine datasets conclude, that XGBoost and the parallel implementation of random forest perform better than more traditional classifiers such as logistic regression on multiple datasets for all evaluated dataset sizes. This disparity increases with the number of observations. Fernández-Delgado et al. (2014) also rate random forest variants as the best performing classifiers for both high and low number of observations. However, in their study, they are closely followed by support vector machines implemented in C.

3. **Number of Predictors** In addition to the number of observations, Fernández-Delgado et al. (2014) also examine the impact of the number of predictors on accuracy. Again, random forests outperform other models, with four different random forest specifications among the top five performing models. Unlike in the other cases, the difference to the C implementation of support vector machines, which is ranked 8th, is greater. In another study, Kirasich et al. (2018) show that above a certain number of predictors, an additional

number of predictors has little impact on the performance of random forests, as opposed to the second model they study, logistic regression.

4. **Noise and Variance among Predictors** In the context of predictors, Kirasich et al. (2018) also examine the information density of the predictors by analysing the effects of noise and variance for logistic regression and random forests using performance measures based on true and false positive rates. They find that with a greater variance among these noise variables and explanatory variables, logistic regression has a higher accuracy than random forests. The reason for this result is the higher false positive rate of random forests, which reduces its accuracy.

5. **Number of Classes** Fernández-Delgado et al. (2014) also take into account the number of classes of the target variable. Their results show that support vector machines (implemented in C) and random forests (implemented in R using caret) perform best on data sets with a larger number of classes. However, this result is less relevant for us, as we are examining the binary case at the other end of the spectrum.

6. **Class Imbalance** Zheng & Jin (2020) go even one step further and also consider the distribution of the different classes. Their analysis shows that while ensemble models like parallel random forests or XGBoost outperform the baseline models such as logistic regression, their stability regarding different degrees of skew is lower.

3.2 Relationship between Model Choice and Subgroup Fairness under Distribution Shift

As we have seen in the previous section, research about the relation between the model choice and subgroup fairness is rare. When focusing on this relation specifically under distribution shifts, this is even more so and no directly related work could be identified[3].

Again, when lifting the constraint of a focus on subgroup fairness, we can identify some works, that have empirically studied classifiers under distribution shifts. However, most studies have been produced in the area of NLP and image recognition, for example by Maron et al. (2021) or Taori et al. (2020). The development of methods that behave fairly also under distribution shift is a second focus of research activities. Wang et al. (2022) give a brief introduction into the topic in their 'Related Work' section.

[3] The literature review method for this survey of related work is documented in A.1.

4 Methodology

Now that we have defined all relevant terms and have examined previous research, we want to describe the methodology we have chosen to study the role of model choice on subgroup fairness and the change of this role under distribution shifts. We start with our dataset and explain how we draw samples from it based on the explanation of distribution shifts in Section 2.4. We then introduce the models we have chosen for comparison and describe them theoretically. Besides the formal setup, we want to highlight theoretical model properties that might influence the performance of the subgroups and shortly mention the chosen implementation method in R. Lastly, we outline the analysis method.

4.1 Dataset

The UCI Adult dataset consists of 32.561 observations and 15 variables and is based on the 1994 US Census database (Kohavi & Becker, 1994). *Sex*, *age* and *gender* were treated as sensitive variables that define 29 subgroups. The binary outcome variable is *income*, which indicates whether the income is either greater than or less than/equal to 50.000 USD. A descriptive analysis of its variables can be found in the appendix under A.2.1.

We performed a two step pre-processing transformation before building the model. First, categorical variables consisting of multiple categories were combined whenever possible to obtain larger groups. This applies to the variables *race*, *working class*, *education*, *marital status*, *occupation* and *country*. Missing values in the columns *workclass* and *occupation* form a new category called *Other*. The variables *capital gain* and *capital loss* were merged into a new variable called *capital change*. This data transformation process was inspired by the work of Nguyen (2017). In the second step, the data was encoded in the dummy format using reference coding and potentially problematic predictors, for example collinear ones, were removed as suggested by M. Kuhn (2019). The variables that finally are used for each distribution shift can be found in the appendix under A.3.

4.2 Creation of Training Samples under Distribution Shift

We use the UCI Adult dataset to create four training samples. One should follow the same distribution as the dataset and will later serve as a baseline to evaluate models regarding subgroup fairness. Three additional training samples should represent different degrees of distribution shifts. As pointed out in Section 2.4, we simulate sample selection bias when generating the distribution shifts. To do so, we assign to each subgroup its size as a weight which determines its sampling probability.

Figure 2 illustrates the two-step sampling process. In the first step, the dataset is randomly split into the training data pool and the test data by the ratio 0.75/0.25 (training data/test data). The motivation behind this split is to avoid overlaps between the test sample and the training samples. In a second step, 60% of the observations of the training data pool are drawn without replacement for each training sample. The first sample is drawn without weights (unshifted) which means that regardless of the subgroup membership, all individuals have an equal probability to become part of the sample. For the three shifted training samples, weights were used and made progressively more extreme by taking the weights to the power 1, 2, and 10. As a result, individuals from the largest group become more and more likely part of the sample.

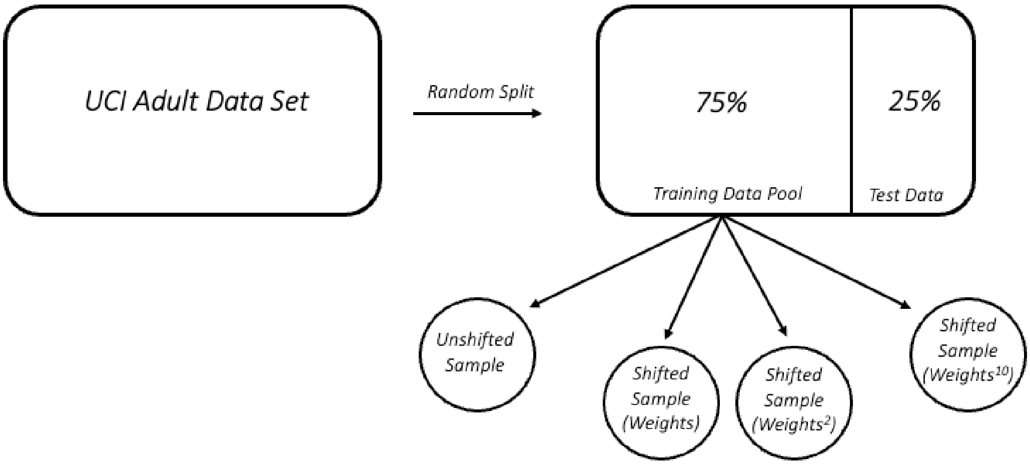

Figure 2: Illustration of the two-level sampling process.

Comparing Figure 4 to Figure 3 illustrates how the sampling bias leads to a sharp increase in the group of white male adults aged 20-60 and a decrease or even disappearance of other groups when using the weights exponentiated by 10.

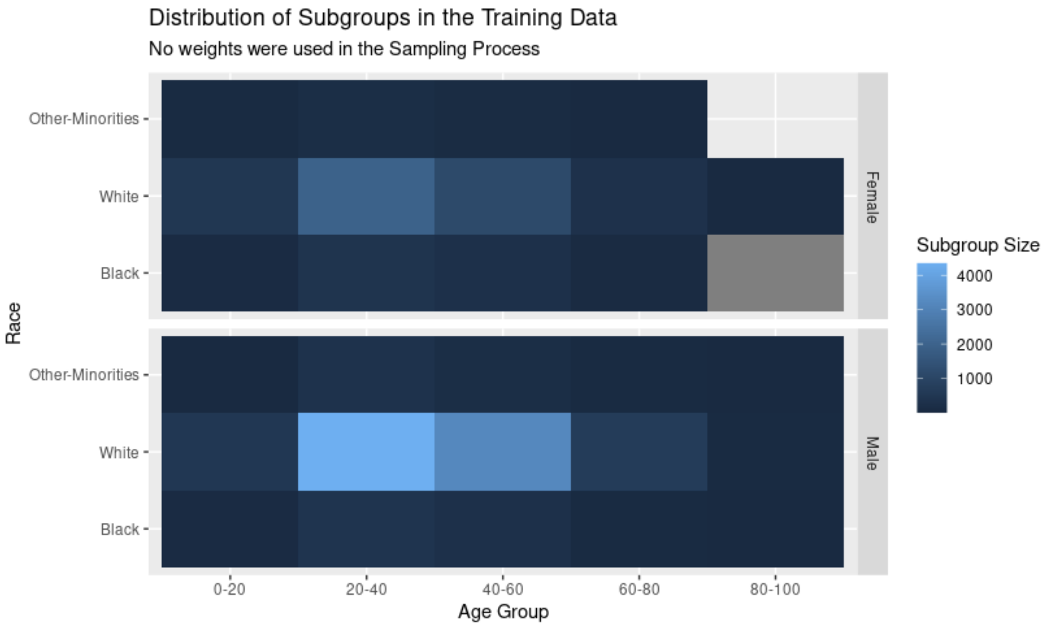

Figure 3: Group Size per Subgroup (Training Data, *No Weights*). The grey area indicates that no individuals from these subgroups are in the sample.

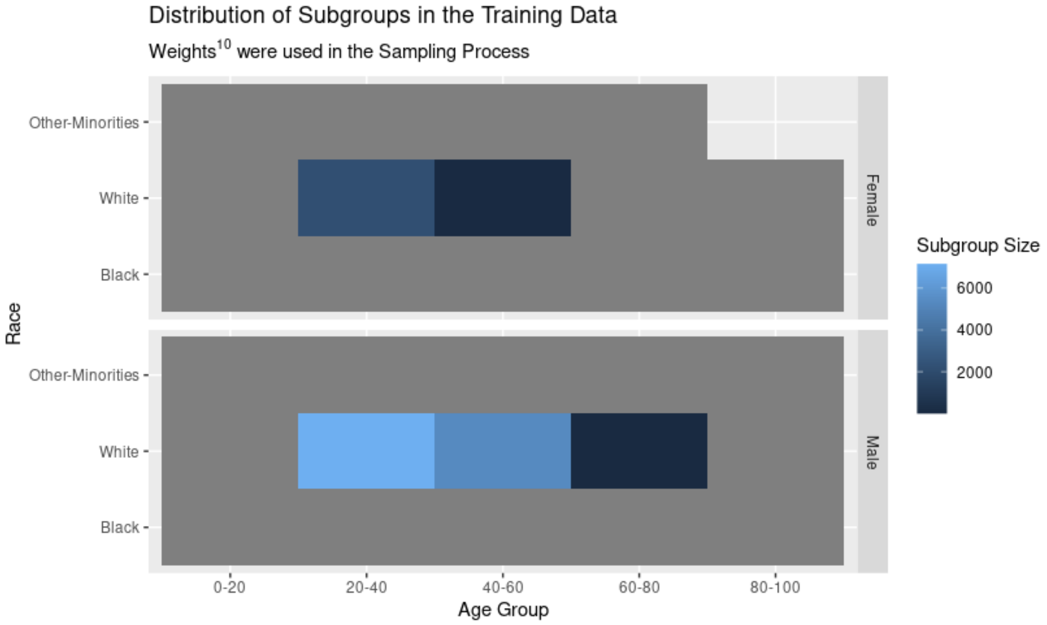

Figure 4: Group Size per Subgroup (Training Data, *Weights*[10]). The grey areas indicate that no individuals from these subgroups are in the sample.

4.3 Model Definitions and Properties

Four classifiers representing different modelling approaches were selected and each of them was trained on the four different training samples. Logistic regression, random forests and XGBoost have already been studied by Bono et al. (2021) while Gardner et al. (2022) investigated tree-based methods and support vector machines among other more specialised models that will not be dealt with here.

4.3.1 Logistic Regression

According to Fahrmeir et al. (2021), the basic idea of regression analysis is to model the expected value of the target variable y[4] as a transformation of a linear combination of variables $x_1, ..., x_k, k \in \mathbf{N}$. For binary variables, we know that $E(y_i) = P(y_i = 1) = \pi_i$, $i = 1, ..., n$. Therefore, we can describe the relationship for them as

$$E(y_i) = \pi_i = h(\beta_0 + \beta_1 * x_{i1} + \ldots + \beta_k * x_{ik}) \tag{4.1}$$

or

$$g(\pi_i) = \beta_0 + \beta_1 * x_{i1} + \ldots + \beta_k * x_{ik} \tag{4.2}$$

where g is the inverse of h. The transformation h is a strictly monotonically increasing cumulative distribution function and guarantees that $h(\beta_0 + \beta_1 * x_{i1} + ... + \beta_k * x_{ik})$ lies between 0 and 1, which is necessary because π_i is a probability. Often, the logistic response function $g(\pi_i) = log(\frac{\pi_i}{1-\pi_i})$ is chosen as transformation in the case of binary variables which is also done here. After transforming both sides of the equation with the exponential function, the model can therefore be described by Equation 4.3, which is generally easier to interpret than working with its non-transformed version.

$$\frac{\pi}{1-\pi} = e^{\beta_0} * e^{\beta_1 * x_1} * \ldots * e^{\beta_k * x_k} \tag{4.3}$$

The parameters $\beta_0, ..., \beta_k$ are unknown and have to be estimated. This is done with the help of maximum likelihood estimation. The covariates can be recorded metrically, binarily or also multicategorically (Fahrmeir et al., 2021). However, the inclusion of categorical variables requires appropriate coding. There are various possibilities for this. For the later analyses, the reference coding described in Fahrmeir et al. (2009) was chosen. The target variable y is assumed to be conditionally independent of the variables x (Fahrmeir et al., 2021).

There are two practical properties that can be derived from this model setup. First, regression models and therefore also logistic regression models can yield unstable predictions when some predictors are (nearly) linear dependent (Fahrmeir et al., 2021). Courvoisier et al. (2011) confirm this behaviour practically. Second, it only has $k + 1$ parameters (Owen, 2007). Therefore,

[4] We adopt the notation of random variables and realisations of such variables from Fahrmeir et al. (2021).

it is less flexible than the subsequently following models in its form specified above.

For our analyses, the logistic regression model was estimated using the R basis function glm() with the binomial distribution via the caret framework.

4.3.2 Random Forests

Decision trees[5] show little bias and work well for complex data as Hastie et al. (2017) explain. However, they have a relatively high variance. Therefore, conducting multiple estimations and then combining the results can greatly improve their performance. This is the underlying idea of the random forest technique, that constructs multiple decision trees from different subsets of the predictors and then combines their individual predictions.

The estimation algorithm for a random forest classifier consists of several steps that are repeated for each tree b of the random forest tree ensemble B (Hastie et al., 2017).

In the first step of building a random-forest tree b, a bootstrap sample containing N observations is drawn from the training data, as Hastie et al. (2017) outlines. From this sample, a random forest tree is built by a recursive strategy. Starting from the root, which contains all observations from the bootstrap sample, m variables are randomly drawn from all available variables. Among the m variables and their values, the best split point is determined using the usual decision tree approach. This involves iterating over all possible binary split points. The set of possible split points is defined by the variables and the values they can assume. The best split is the split that leads to the largest improvement in purity, or in other words, the best improvement in separating the two classes of target variables in the two daughter nodes. This is defined by an impurity measure, in most cases the Gini Coefficient which in the binary case takes the form $2p(1-p)$ with class proportion p. The greatest improvement in purity can therefore be reached by minimising the following expression with respect to the optimal split variable j and split point s among the m randomly drawn variables

$$min_{j,s}(n_1 * 2p_{1,j,s}(1-p_{1,j,s}) + n_2 * 2p_{2,j,s}(1-p_{2,j,s})) \tag{4.4}$$

where n_1 and n_2 denote the number of observations in the two daughter nodes. The preceding steps are then applied to the resulting daughter nodes until the minimum number of observations n_{min} per node is reached.

From the resulting random forest trees and their individual predictions for a new observation, we obtain the aggregate prediction by a so-called "majority vote" of all trees, i.e. the new observation is assigned to the class that is predicted by the majority of the trees (Hastie et al.,

[5] For more information about decision trees, please consult Hastie et al. (2017), Chapter 9.2.

2017, p. 588).

In practice, random forests show strong performances on many datasets. However, this behaviour has not yet been fully understood from a theoretical side, as Denil et al. (2014) state. Therefore, multiple empirical studies combined with theoretical thoughts have been undertaken. Grinsztajn et al. (2022) identify two important factors in their analysis for tabular data. First, random forests might have an edge over other classifiers because they can learn irregular patterns. This is because the underlying trees can construct piece-wise constant functions. Therefore, they can initially perfectly fit every single data point, as Wyner et al. (2017) point out. They argue that this has a positive impact on the performance as it reduces the influence of random noise on the overall classifier. Such points will only impact their very local neighbourhood. The process of averaging at the same time prevents overfitting. The second reason that Grinsztajn et al. (2022) put forward is that random forests are hardly affected by uninformative features that according to the authors frequently appear in tabular data, by the possibility of decreasing feature importance during the training process. However, it is important to notice that their analysis has only been conducted with numerical data. Unlike logistic regression, random forests are also less affected by multicollinear predictor variables since the trees are constructed in a decorrelated manner by only considering a random selection of variables per tree (Hastie et al., 2017).

For our analyses, the random forest model was trained using the R package randomForest via the caret framework. This setting allows tuning the number of randomly selected predictors (M. Kuhn, 2019).

4.3.3 Support Vector Machines

Support vector machines can be used to non-linearly classify observations based on a multidimensional decision boundary (Hastie et al., 2017; Murty & Raghava, 2016). This spatial boundary is only determined by the observations that lie near the intersection between the two classes, making this approach less prone to outliers.

The underlying idea of support vector machines is to construct a hyperplane that optimally separates the two classes -1 and 1 (Hastie et al., 2017). A hyperplane in $\mathbb{R}^n$ is defined as a set P that consists of all $x \in \mathbb{R}^n$ that satisfy $< x, u >= c$, where c is a constant and $u \neq 0$ (Binmore, 1981). Therefore, a hyperplane can intuitively be viewed as a subspace with one less dimension than the vector space such as a two-dimensional layer in a three-dimensional space (Weisstein, n.d.).

Based on this definition of a hyperplane, a binary linear classification problem can be described by the separating hyperplane

$$\{x : f(x) = x^T * \beta + \beta_0 = 0\} \tag{4.5}$$

with β being a unit vector and x defined as before (Hastie et al., 2017). One would then classify the observations x based on the sign of the value of $f(x)$. Hastie et al. (2017) now extend this, step by step, to the non-linear situation. The aim is to maximise the distance between the observations of the two classes and the hyperplane, the so-called margin M, in order to achieve optimal classification performance on unknown data. Since classes may overlap, we maximise the margin M, but tolerate that some points violate it. The function $f(x) = x^T * \beta + \beta_0 = 0$ gives the signed distance of an observation x from the hyperplane and is positive when multiplied by the class membership $y_i \in \{-1, 1\}$, i = 1, ..., n in the case of correct classification. Therefore, M is maximised under constraint 4.6 where ξ_i is a so-called slack variable given by $\forall i,\ \xi_i \geq 0,\ \sum_{i=1}^{N} \xi_i \leq$ constant . A slack variable generally indicates "how much a constraint is violated" (Ernst & Schweikard, 2020, p. 34).

$$y_i(x_i^T * \beta + \beta_0) \geq M - \xi_i \tag{4.6}$$

After a few transformations[6], the optimisation problem can be expressed using Lagrange multipliers α_i and μ_i:

$$L_p = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i[y_i(x_i^T\beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^{N} \mu_i\xi_i \tag{4.7}$$

This expression is minimised regarding β, β_0 and ξ_i (Hastie et al., 2017). The cost parameter C indicates how "costly" it is when the constraint is violated (Batuwita & Palade, 2013). By determining the respective derivatives, re-substituting them into the function and considering the Karush-Kuhn-Tucker conditions that guarantee the optimality of the solution according to H. W. Kuhn & Tucker (1951), we obtain the estimator (Hastie et al., 2017):

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i \tag{4.8}$$

with the Lagrange multiplier $\hat{\alpha}_i$ being non zero for some observations only. The Karush-Kuhn-Tucker conditions imply that this is the case for observations that are either at the margin or within the margin. The latter case can occur either due to a classification error or a so-called boundary violation. Therefore, the classifier does not depend on all observations. The observations on which it depends are called 'support vectors'.

[6] This explanation summarises only the main steps in the construction of the Support Vector Machine. For more information, please read Hastie et al. (2017) Chapters 4.5, 12.2, 12.3

This technique can be generalised for nonlinear separation problems by using kernels which is referred to as support vector machine (Hastie et al., 2017). The main idea of these kernels is to project the feature values x_i into a space with higher dimension using a function $h_l(x),\ m = 1, ...\ L$. Consequently, the Lagrangian optimisation problem (see 4.7) can be reformulated to include the input features only as inner products $\langle h(x_i), h(x_j^)\rangle$, which is called the kernel function $K(x_i, x_k)$. There are several possibilities for the kernel function, but it should be symmetric and positive (semi-) definite (see 4.10). As in the linear case, the observations are classified by:

$$\hat{G}(x) = sign(\hat{f}(x)) \tag{4.9}$$

In practice, one main advantage of support vector machines is that they depend only on the observations at the margin (Hastie et al., 2017). As mentioned in the introductory paragraph, this model property makes them less prone to outliers. Besides robustness to extreme values, another important behaviour in practice is the sensitivity to class imbalance. Batuwita & Palade (2013) point out that the same misclassification cost C is assumed for both classes. When the one class is a lot smaller than the other class, observations from this class may get disproportionally misclassified, as they explain. However, they also mention that the performance on "moderately skewed" datasets is still acceptable, probably due to an internal correction mechanism for moderately imbalanced datasets that is induced by the Karush-Kuhn Tucker constraints (Batuwita & Palade, 2013, p. 5). The relevant constraint $0 = \sum_{i=1}^{N} \alpha_i y_i$ is obtained from the Lagrange equation 4.7 (Hastie et al., 2017). Batuwita & Palade (2013) point out that in a case where one class outnumbers the other class, the minority class must have higher values for the coefficients α_i to still satisfy the zero-sum condition. Since the coefficients serve as weights for the classification decision, the initial imbalance is being corrected.

For the computational implementation of support vector machines in R, the package kernlab was chosen since it offers a wide range of kernels. Among these, the Gaussian radial basis function

$$k(x, x^T) = exp(-\sigma||x - x^T||^2),\ \sigma > 0 \tag{4.10}$$

was selected because there is no prior knowledge about the data structure (Karatzoglou et al., 2004). Additionally, it is seen as a good start by researchers such as Hsu et al. (2003). Like for the other models, the implementation was conducted with the caret package that tunes the sigma parameter of the kernel function as well as the cost of misclassification (M. Kuhn, 2019).

4.3.4 XGBoost

The XGBoost algorithm is based on the concept of gradient tree boosting, but is more efficient from a computational point of view (Chen & Guestrin, 2016).

The XGBoost classifier $\hat{y}_i$ uses an ensemble of regression trees[7] f_k of which each returns prediction scores $w \in \mathbb{R}^T$ (T number of leaves) as Chen & Guestrin (2016) explain. They complement each other by adding their individual predictions, which can be seen in the following classifier formula

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{4.11}$$

where m denotes the number of features and $x_i \in \mathbb{R}^m$. The functions f_k are determined by minimising the objective $L(\phi)$ in 4.12, where $l(\hat{y}_i, y_i)$ is a differentiable loss function considering the difference between prediction and actual value $y_i \in \mathbb{R}$ and where $\Omega(f) = \gamma * T + \frac{1}{2} * \lambda||w||^2$ is a penalty term that extends the gradient tree boosting approach and reduces overfitting.

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{4.12}$$

The parameters of the model that we want to optimise are the individual tree structures f_k. Since they cannot be calculated simultaneously, they are determined additively. In a step t, the f_t is chosen so that

$$L^t = \sum_i l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t) \tag{4.13}$$

is optimised. The initial predictive value $\hat{y}_i$ has to be manually defined and is set to 0.5 per default in their computational implementation (Chen et al., 2022). To put Equation 4.13 into a more computationally practical form, the Taylor expansion of the loss function is taken up to the second order (Chen & Guestrin, 2016). Omitting the constants and inserting the optimal value for w_j^*, we obtain 4.14, which can be regarded as the "quality" for the tree structure q

$$\tilde{L}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{4.14}$$

where g_i and h_i are the first and second order derivatives of the loss function (Chen & Guestrin, 2016, p. 3). This results in

$$L_{split} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I_I} g_i)^2}{\sum_{i \in I_I} h_i + \lambda} \right] - \gamma \tag{4.15}$$

as a criterion for selecting each split for the tree f_t, where I_l and I_r are the instance sets of the two nodes resulting from the split, since the computation of 4.14 for all possible trees is practically unmanageable (Chen & Guestrin, 2016). It returns the loss reduction induced by the split. Generally, this additive approach is a pragmatic one and does not necessarily give the optimal solution (XGBoost documentation, 2022).

[7] For more information on regression trees, see Hastie et al. (2017) Chapter 9.2.2.

To make the algorithm computationally more efficient, Chen & Guestrin (2016) propose three changes in addition to introducing the penalty term $\Omega(f_k)$.

First, they propose shrinkage and column subsampling to further reduce overfitting. The idea of shrinkage is to reduce the scores predicted by the leaves, also called leaf weights, by a factor a. Column subsampling uses only a sample of randomly drawn variables to determine each split, which further increases computational speed and prevents overfitting.

Secondly, they introduce two adaptations to the splitting technique that are more efficient. The first is an approximate splitting algorithm, which essentially consists of providing the percentiles of the feature as suggestions for potential splits. These suggestions can either be made at the beginning of each tree construction or updated after each split, which may be more useful for deeper trees. In addition, they increase the splitting speed by taking into account the frequency of sparse data in real-world applications. Assigning a default direction in each branch learned from the data, increases speed by a factor of 50.

Third, they propose three adjustments to the system design: Column block for parallel learning, cache-aware access and blocks for out-of-core computation, which are not discussed further here as they do not affect the model result but only the computational speed.

In practice, XGBoost shows similar properties as those outlined by Grinsztajn et al. (2022) in the chapter about random forests. In summary, they explain how the tree structure leads to a greater robustness against noisy observations as well as robustness against uninformative features which has a positive impact on the overall performance. Additionally, XGBoost penalises uninformative leafs as Nielsen (2016) points out, which makes it even more flexible. The Newton boosting method and the relatively high-order approximation give it an additional competitive edge, also over other boosting approaches.

For the computational implementation of XGBoost in this work, the tree learning algorithm of the xgboost package was chosen and like the other models implemented in caret (M. Kuhn, 2019). In this settings, the number of boosting iterations, maximum tree depth, shrinkage, minimum loss reduction, subsample ratio of columns, minimum sum of instance weights and subsample percentage are optimised as hyperparameters (M. Kuhn, 2019).

4.4 Subgroup Fairness Analysis

4.4.1 Selected Performance Measures

As defined in Section 2.3, our definition of subgroup fairness is based on performance parity of underlying performance measures. Ferri et al. (2009) divide performance measures into three different groups: threshold-based measures, when minimising the total number of misclassifications is of interest, probabilistic measures, when the "deviation from the true probability" should be minimised, and rank-based measures, which are useful when the goal is to "select the best n instances", such as for recommendations (Ferri et al., 2009, p. 1-2). As we want to measure the number of errors per subgroup the first group of measures is relevant for us. Among those, two were chosen to measure the prediction quality in each group: the Accuracy and the Matthews Correlation Coefficient (MCC).

The Accuracy of a classifier for N individuals is defined as

$$Acc = \frac{TP + TN}{N} \tag{4.16}$$

where TP is the number of true positives and TN is the number of true negatives, based on the notation of Fernández et al. (2018). The accuracy measure was chosen because of its intuitive interpretation as classification error. However, it has a major drawback, as Fernández et al. (2018) explain. Depending on the skew of the data, high accuracy can be achieved without actually obtaining a good classifier. An example would be the prediction of rare events that occur with very small probabilities of for example 1%, as they illustrate. If the non-occurrence of the event is predicted for all observations, this would result in a classifier with an accuracy of 99%, but useless for predicting the event of interest.

Descriptive analysis of the data shows that in our case, the outcome variable $income > 50.000$ USD is highly imbalanced within some groups (see Figure 7). For such cases, Fernández et al. (2018) suggest the use of Cohen's kappa, MCC, F-measure or its variation G-measure. Since the F-measure focuses on the positive class, while in the income situation, we consider both classes as equally important, and Cohen's Kappa can return higher scores for classifiers that actually perform worse in some situations, MCC was chosen as a second measure (Delgado & Tibau, 2019). It is specified by

$$MCC = \frac{TP * TN + FP * FN}{\sqrt{POS * NEG * PPOS * PNEG}} \tag{4.17}$$

where POS is the total number of individuals in the positive class, NEG is the total number of individuals in the negative class, FP is the total number of false positives and FN is the total number of false negatives (Fernández et al., 2018). $PPOS$ specifies the number of predicted individuals in the positive class and $PNEG$ is the number of predicted individuals in the negative class. The MCC is conceptually related to the Pearson correlation coefficient and the chi-squared contingency table and considers all four parts of a confusion matrix. It takes values ranging from -1 to 1, where -1 indicates that the classification is always wrong, 0 that it is no better than chance and 1 that it is always right.

4.4.2 Analysis of Subgroup Fairness

For our analysis, we need to calculate for each subgroup, model and distribution shift the Acc and MCC performance measure. We then divide our analysis in two steps. In the first step, we analyse the role of the model choice for subgroup fairness. This is done on the unshifted sample to rule out the effects of distribution shifts. To analyse the role of the model choice on subgroup fairness, we first determine whether the subgroup fairness differs between classifiers. We do so by computing the two subgroup fairness criteria from Section 2.3. To better understand why or why not the subgroup fairness differs across classifiers, we then zoom in and compare the classifiers on a subgroup level. In the second step, we analyse if and how the relationship between the model choice and subgroup fairness changes under distribution shift. We proceed analogously. First, we calculate our subgroup fairness measure for all classifiers and compare the values of the models across the shifts. We then also move to the subgroup level to better understand, why or why not subgroup fairness differs across classifiers and shifts.

5 Results

The models presented in Section 4 were implemented in R using the caret package. Based on Bergstra & Bengio (2012)'s findings, the hyperparameters are optimised using a random search that consists of "independent draws from a uniform density from the same configuration space as would be spanned by a regular grid" (Bergstra & Bengio, 2012, p. 3). This approach has proven to be more efficient, especially in cases where there is no prior knowledge about the optimal hyperparameter range (Bergstra & Bengio, 2012). The number of hyperparameter combinations evaluated is set to 50. The hyperparameters are then estimated using a 10-fold cross validation with five repetitions, as recommended by M. Kuhn & Johnson (2019) for smaller datasets. To allow comparisons between the models, the same variables are used for all models (see A.3). Their subgroup fairness will now be analysed based on the structure outlined in Section 4.4.2.

5.1 Relationship between Model Choice and Subgroup Fairness

First, we are interested in whether the model choice has an influence on subgroup fairness at all. Therefore, we determine the classifiers' individual subgroup fairness for the unshifted case by computing the Subgroup Performance Parity criteria.

Results on Aggregated Level *Criterion 1* measures the maximum difference that occured between two subgroups. Since we are only interested in relative comparisons between the classifiers, we do not set a value for α here. When selecting the Acc as performance measure, the maximum accuracy difference is around 0.3 for all classifiers, as Table 1 shows. Random forests hold a maximum difference that is slightly smaller than for the others. On the other hand, when selecting the MCC as a performance measure, the variation between classifiers is higher as can be seen in Table 2. It has to be taken into account that MCC in general has a broader value range than Acc. Here, support vector machines produce the smallest maximum difference.

Maximum Performance Difference between Subgroups				
Distribution Shift	*unweighted*	*exponent 1*	*exponent 2*	*exponent 10*
Logistic Regression	0.311	0.313	0.312	0.312
Random Forests	0.302	0.311	0.312	0.308
Support Vector Machines	0.310	0.327	0.321	0.311
XGBoost	0.310	0.317	0.308	0.309

Table 1: Criterion 1 for Subgroup Performance Parity (Acc)

Maximum Performance Difference between Subgroups				
Distribution Shift	*unweighted*	*exponent 1*	*exponent 2*	*exponent 10*
Logistic Regression	0.763	0.763	0.785	0.802
Random Forests	0.712	0.763	0.750	0.763
Support Vector Machines	0.627	0.712	0.723	0.763
XGBoost	0.670	0.888	0.763	0.763

Table 2: Criterion 1 for Subgroup Performance Parity (MCC)

Criterion 2 measures the variation of performance values between subgroups. As we are only interested in relative comparisons between the classifiers, we once again do not set a value for ϵ. When selecting the Acc as performance measure, we observe a similar range of values for all classifiers (see Table 3). In this case, support vector machines perform better than random forests, however only with a small lead. When looking at the MCC in Table 4, the values also are close to each other, especially taking into account that the MCC is defined on a broader interval than the accuracy. As for the Acc, support vector machines perform best.

Acc Standard Deviation				
Distribution Shift	*unweighted*	*exponent 1*	*exponent 2*	*exponent 10*
Logistic Regression	0.1081	0.1098	0.1132	0.1134
Random Forests	0.1092	0.1080	0.1129	0.1077
Support Vector Machines	0.1077	0.1083	0.1046	0.1057
XGBoost	0.1108	0.1103	0.1089	0.1081

Table 3: Criterion 2 for Subgroup Performance Parity (Acc)

MCC Standard Deviation				
Distribution Shift	*unweighted*	*exponent 1*	*exponent 2*	*exponent 10*
Logistic Regression	0.2346	0.2243	0.2269	0.2349
Random Forests	0.2209	0.2275	0.2176	0.2335
Support Vector Machines	0.1965	0.2148	0.2312	0.2356
XGBoost	0.2178	0.2304	0.2262	0.2268

Table 4: Criterion 2 for Subgroup Performance Parity (MCC)

Combining both criteria, support vector machines are more subgroup-fair for the MCC as a underlying performance measure but only by a small margin. For the Acc, we can not identify one classifier to be fairer than the others. These results overall still question the importance of the classifier. Such similar values could either occur because all classifiers show very similar performance per subgroup or show similar performance values that are distributed differently across the subgroups per classifier. Therefore, we will have a closer look at subgroup level to get a clearer picture about the distribution of performance values.

Zooming in on Subgroups When conducting pairwise comparisons such as in Figure 5 and 6, we observe that the subgroups can be split in two cases. On the one hand are those, for which the performance differences between classifiers are very small, such as white females and males between 20 and 60. And on the other hand are those where the corresponding values do vary by about +/- 0.1 for Acc and +/-0.2 for MCC. However, no clear pattern can be observed. This seems to be plausible when looking at the group size and skew of the target variable (cf. Figure 7 and 8). The subgroups for which the performance varies a lot between classifiers can be characterised by comparably low subgroup sizes which makes small changes appear big. The subgroups for which the performance varies little between classifiers are of larger subgroup sizes and either have a more balanced target variable or one class of the target variable does not exist. Overall, we can not observe a clear indication that the model choice plays a notable role for subgroup fairness.

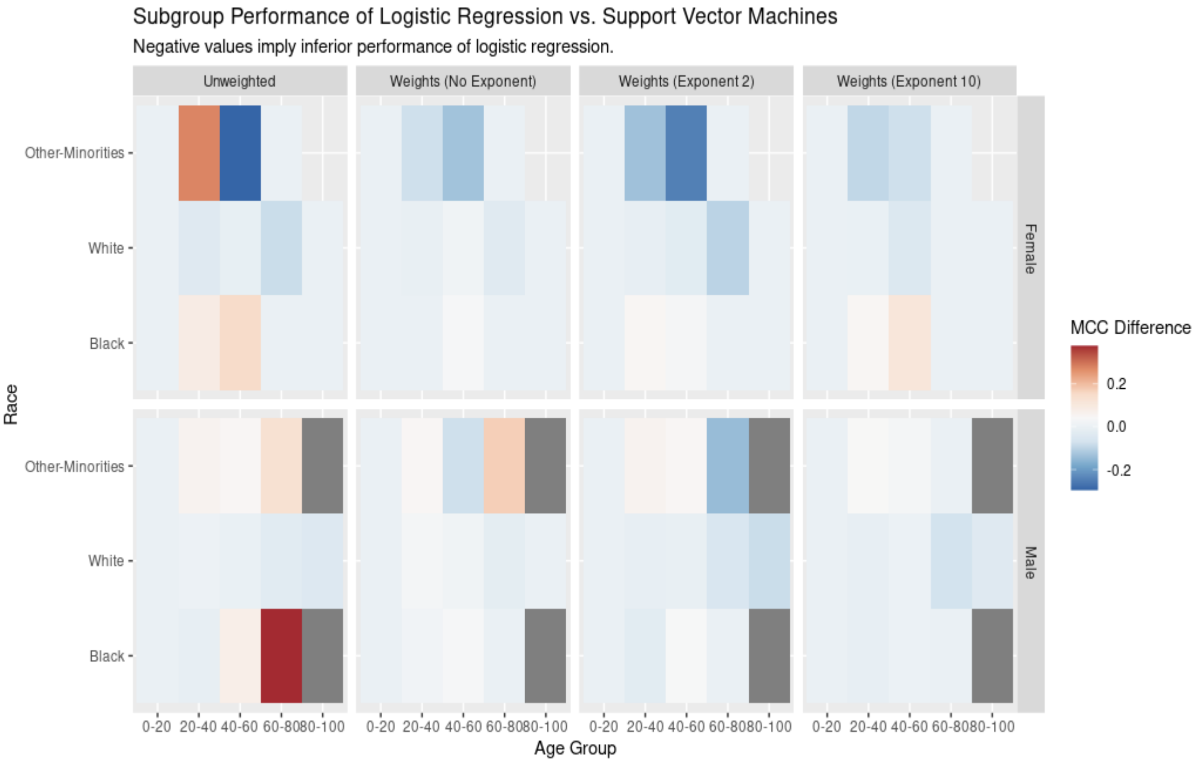

Figure 5: Subgroup Performance Difference between Logistic Regression and Support Vector Machines (MCC). The grey areas indicate that no individuals from these subgroups are in the sample. For all other classifiers, the figures can be found in the appendix.

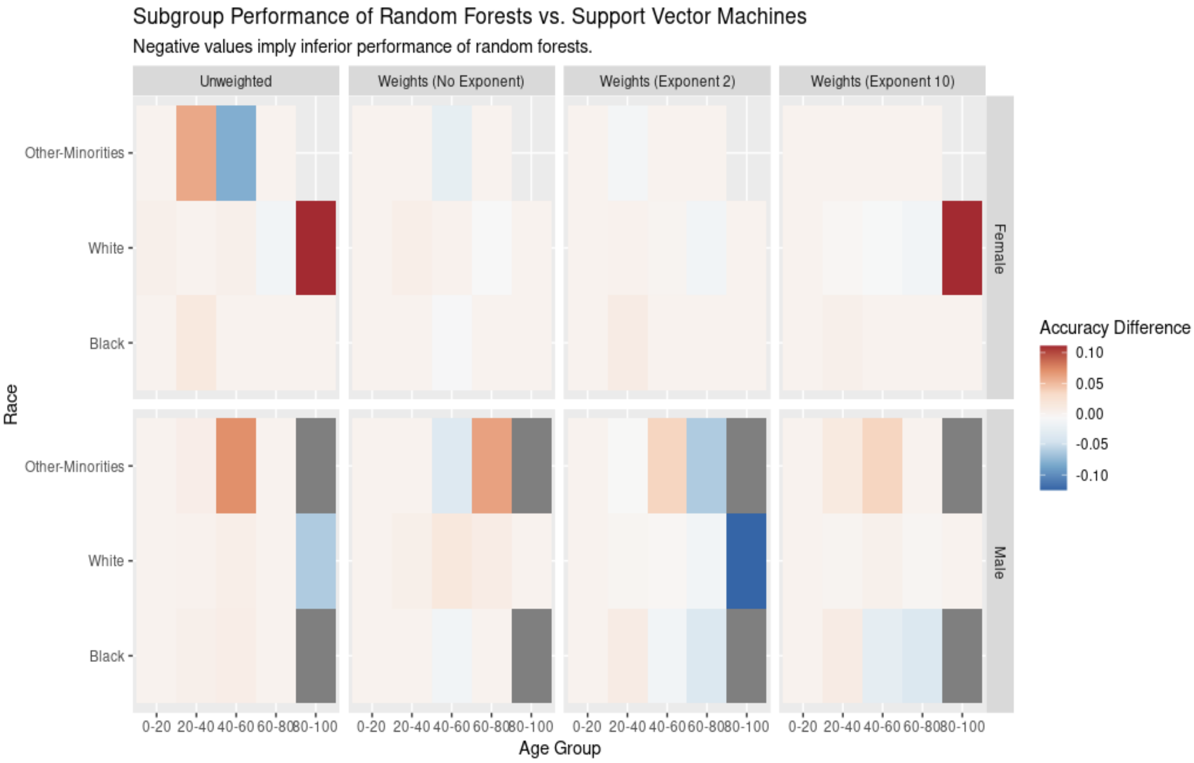

Figure 6: Subgroup Performance Difference between Random Forests and Support Vector Machines. The grey areas indicate that no individuals from these subgroups are in the sample. For all other classifiers, the figures can be found in the appendix.

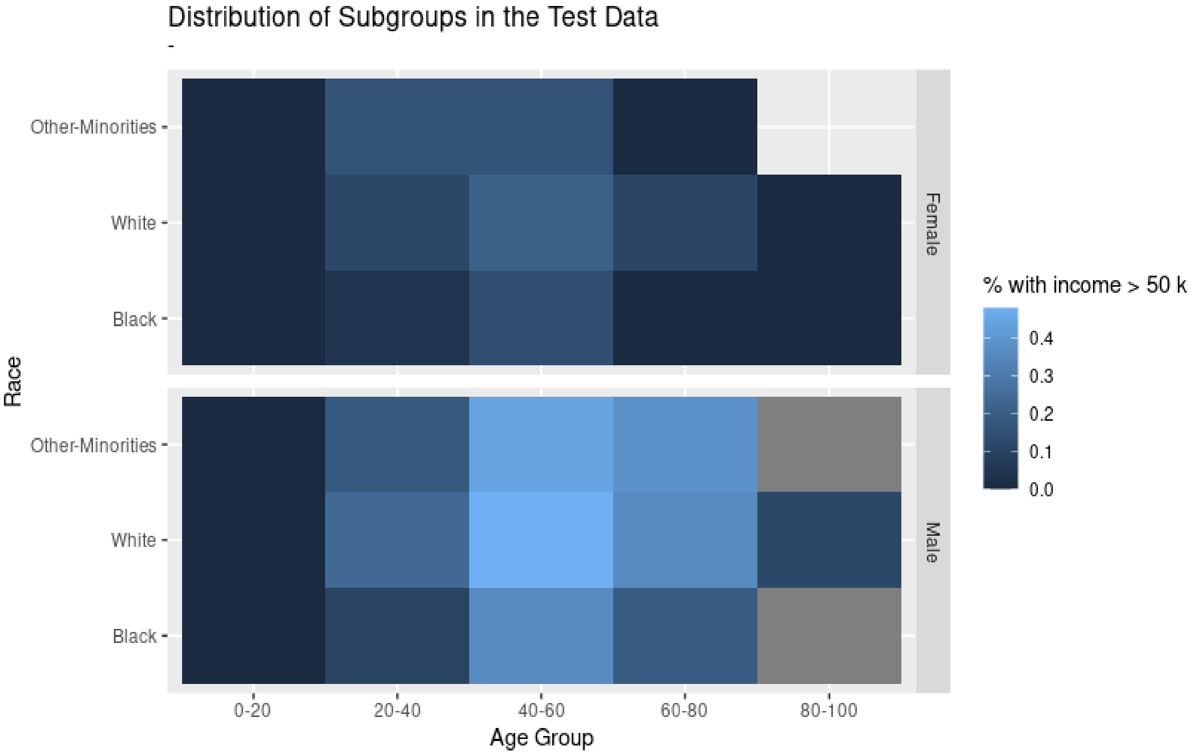

Figure 7: Income Distribution per Subgroup (Test Data). The grey areas indicate that no individuals from these subgroups are in the sample.

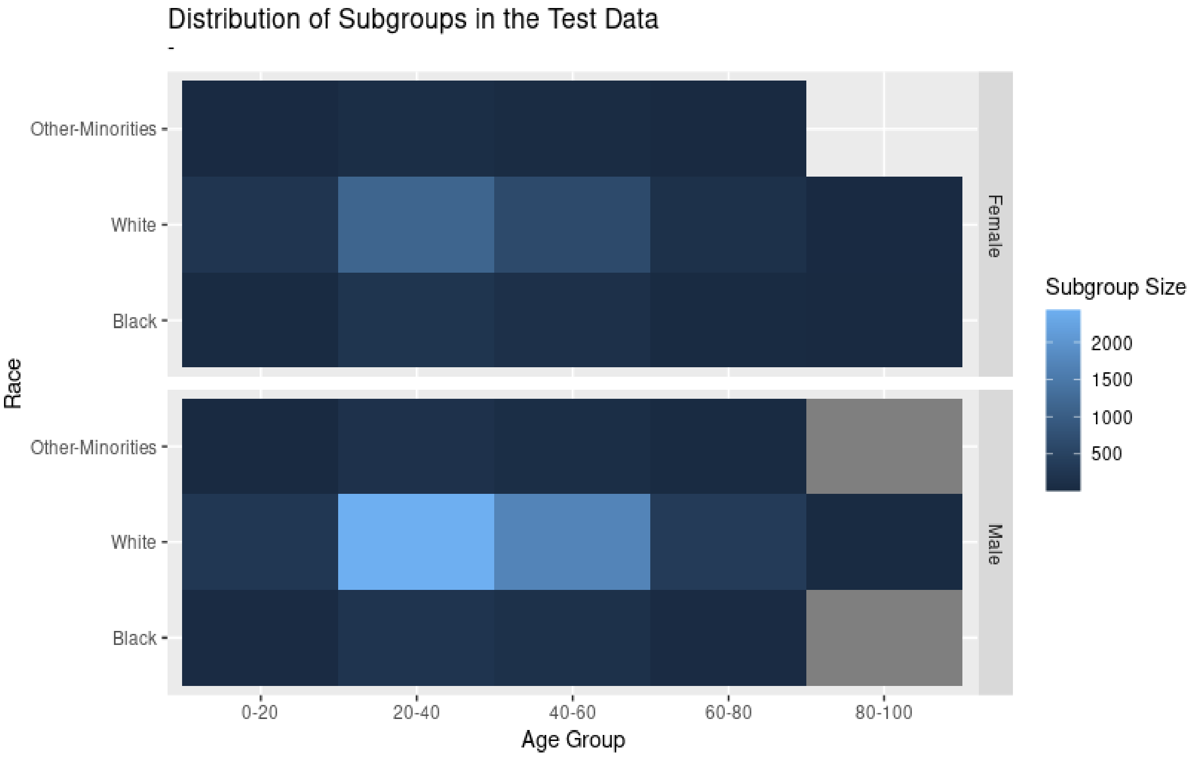

Figure 8: Group Size per Subgroup (Test Data). The grey areas indicate that no individuals from these subgroups are in the sample.

Relation to Previous Research The results described above are partly supported by previous research. Bono et al. (2021) found that subgroup performance disparities are similar for the models they studied which the subgroup fairness analysis has also indicated in our case. Even though the support vector machines performed best, they led only by a small margin. Gardner et al. (2022) observed that random forests generally perform better across subgroups. In our case, random forests showed average performance.

5.2 Relationship between Model Choice and Subgroup Fairness under Distribution Shift

We now want to study whether distribution shifts cause different results than observed in the Section before.

Results on Aggregated Level We therefore compare the classifiers' subgroup fairness values across the shifts and start with *Criterion 1.* When using Acc, for all classifiers the maximum performance difference increases under shift with exponent 1, but then decreases again and overall remains in the same value range. However, in view of the small range this might still be a random pattern. Random forests perform best under shifts with exponent 1 and 10 but are outperformed by XGBoost for exponent 2. Therefore, no classifier is better than the others across all shifts for Criterion 1. For MCC, the values slightly increase with more extreme distribution shifts which means that the models become unfairer. Across all shifts, support vector machines have the best value for Criterion 1. However, the differences to the other classifiers might still be random.

For *Criterion 2*, we again observe that values for all classifiers lie within a similar range. For Acc, there is no clear trend observable across the shifts. Depending on the shift, random forests or support vector machines have the best value. For MCC, the values slightly increase with more extreme distribution shifts which means that the models become unfairer. For each shift, a different classifier performs best (cf. Tables 3, 4).

Combining both criteria, we can not make a clear statement about which classifier is best with respect to subgroup fairness under distribution shifts. We now zoom in on subgroup level to examine whether the differences between the subgroup performances show a similar pattern across distribution shifts as in the unshifted case.

Zooming in on Subgroups As we can see in Figures 6 and 5, the performance difference between large groups remains small, while small groups continue to vary across shifts. Nevertheless, a clear trend can not be identified. The performance differences between the classifiers stay relatively constant for Acc with the exception of the differences between logistic regression and the remaining classifiers that increase, as Table 5 shows. When looking at the MCC, some disparities even decline as can be seen in Table 6. However, this trend might be random as well.

Acc Standard Deviation				
Distribution Shift	unweighted	exponent 1	exponent 2	exponent 10
LR vs. RF	0.0223	0.0421	0.0429	0.0430
LR vs. SVM	0.0374	0.0457	0.0482	0.0479
LR vs. XGB	0.0153	0.0192	0.0441	0.0425
RF vs. SVM	0.0347	0.0159	0.0285	0.0250
RF vs. XGB	0.0116	0.0427	0.0209	0.0046
SVM vs. XGB	0.0329	0.0492	0.0216	0.0246

Table 5: Subgroup Performance per Classifier Comparison (Acc): Standard Deviation; Abbreviations used: LR (Logistic Regression), RF (Random Forest), SVM (Support Vector Machines), XGB (XGBoost)

MCC Standard Deviation				
Distribution Shift	unweighted	exponent 1	exponent 2	exponent 10
LR vs. RF	0.0871	0.0346	0.0688	0.0629
LR vs. SVM	0.1179	0.0510	0.0679	0.0399
LR vs. XGB	0.0469	0.0478	0.0595	0.0499
RF vs. SVM	0.0937	0.0441	0.0559	0.0551
RF vs. XGB	0.0514	0.0357	0.0535	0.0261
SVM vs. XGB	0.0919	0.0695	0.0648	0.0451

Table 6: Subgroup Performance per Classifier Comparison (Matthew's Correlation Coefficient): Standard Deviation; Abbreviations used: LR (Logistic Regression), RF (Random Forest), SVM (Support Vector Machines), XGB (XGBoost)

Further analysis Our definition of subgroup fairness focuses on the relative performance and leaves the absolute performance value untouched. The latter also is important in practice since better performing models generally are more desirable for everyone. We therefore now briefly look at the absolute performance of classifiers. Figure 9 or 10 show that performance remains relatively stable across all shifts. This means that more observations for large groups do not seem to improve the performance of these groups. Likewise, the reduction of individuals from minority groups (see for instance Figure 4 for the distribution of the different training datasets) does not decrease their performance.

As a reaction to these results, we have retrained the models, this time leaving out all variables that had the highest variable importance based on the caret model output. All steps were carried out analogously. The graphical results can be found under A.5 .

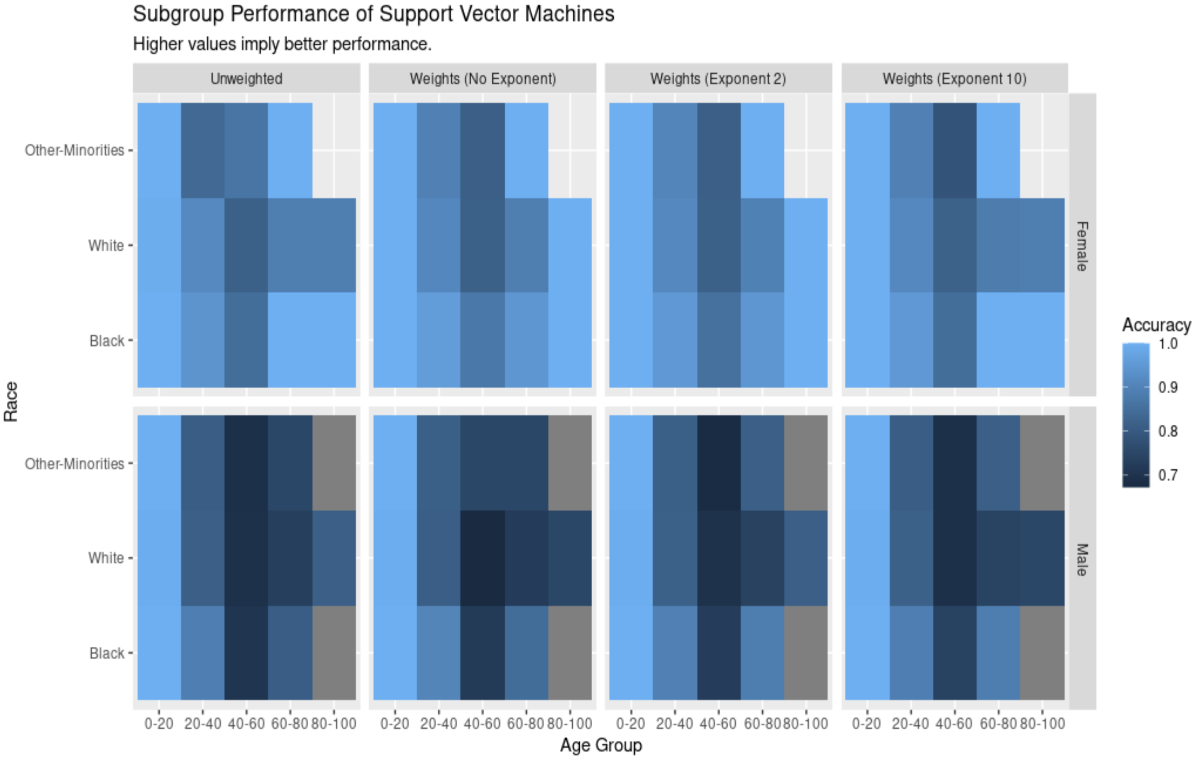

Figure 9: Subgroup Performance of Support Vector Machines (Acc). The grey areas indicate that no individuals from these subgroups are present in the sample.

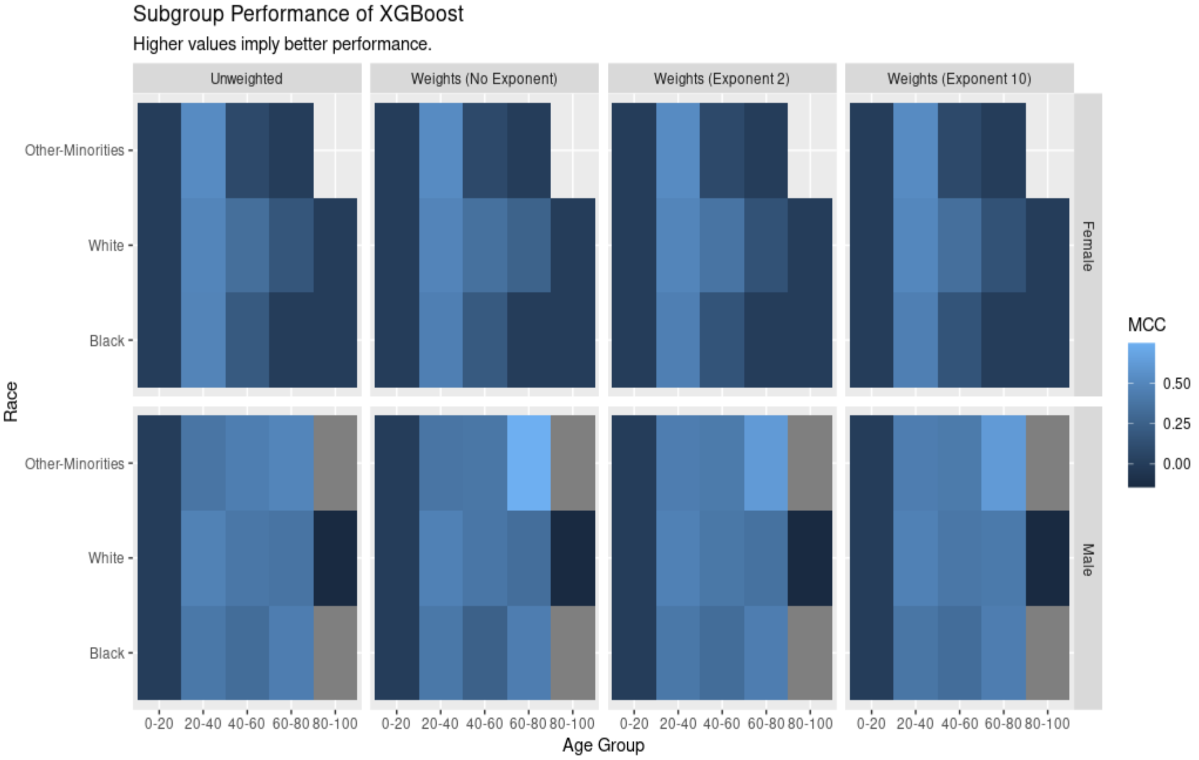

Figure 10: Subgroup Performance of XGBoost (MCC). The grey areas indicate that no individuals from these subgroups are present in the sample.

6 Discussion

This thesis provides a first exploration of the role of model choice on subgroup fairness under different degrees of distribution shifts. Overall, we found that this role is limited on unshifted as well as shifted training samples and only found a slight indication that support vector machines are more subgroup-fair on unshifted data when using the MCC as performance measure.

The UCI Adult dataset was chosen for this analysis because of its high popularity in the fairness community. However, numerous researchers such as Ding et al. (2021) have raised criticism about it. According to them, one major drawback of the UCI Adult dataset is that the threshold of the target variable of 50.000 USD leads to a very imbalanced distribution among the different races. 24% of the white individuals and only 12% of the black individuals earn more than 50.000 USD. Additionally, they point out that minor disadvantages such as a lack of good documentation or a binary gender variable further limit its practical usability. We have also experienced difficulties in practice. Distribution shifts were challenging to create because of very large differences in subgroup sizes.

In order to create samples under distribution shifts, weights have been used in the sampling process. We first have used the ”fnlweight” variable for this purpose. However, this approach has not led to a distribution change among the sensitive variables. As a result we have adopted our own sampling process that involves the scaled group weights and their exponentation. Among a set of possible options, the interval [0.3, 3] and the exponents 1, 2, and 10 were chosen. It remains unclear, whether different exponents or methods to create distribution shifts would lead to different results.

As pointed out at the top of the page, the differences between models and shifts have been small. There are multiple possible reasons for this. On the one hand, classifiers could actually behave similarly on the subgroups which seems unlikely in light of the findings of Delgado & Tibau (2019) for overall performance. On the other hand, this result could have been caused by highly predictive variables in the dataset which work well across subgroups and shifts. This motivates follow-up research using different datasets to uncover the actual reasons and investigate whether the difference that we did find are significant.

Recent research by Lum et al. (2022) has revealed that meta-metrics such as the max-min difference or max-min ratio that aim at capturing disparities between subgroups in one number are often biased and overestimate the level of unfairness. They suggest an alternative measure to compute the variance between subgroup metrics which could not be applied in this work because of the lack of applicability for MCC. Nevertheless, their result shows that subgroup variances in our calculations should not be overestimated.

7 Conclusion

The goal of this thesis was to provide a first exploration of the role of model choice for subgroup fairness on shifted and unshifted data. In the first step, we have formalised our notion of subgroup fairness based on two criteria - max-min difference and standard deviation of the underlying performance measure - applied on subgroup level (*Subgroup Performance Parity*). We then created different training samples from the UCI Adult dataset that were shifted with respect to the subgroup sizes and implemented four typical machine learning models. In the last step, we conducted a descriptive analysis of the subgroup performance results. This explorative analysis has produced two results.

First, the role of the model choice appears to play a limited role in our example. The subgroup fairness values of the classifiers were relatively similar. When using the MCC as performance measure, support vector machines are more subgroup-fair, but only by a small lead. Looking more closely at the individual subgroups, we observed that for larger subgroups models showed equal performance, while more variation occurred for smaller groups which is not surprising.

Second, the role of model choice does not seem to change much between shifted and unshifted data in our specific case. We have seen that subgroup fairness values did not differ much between classifiers. Only for the MCC, we could observe a slight increase in performance variation among the subgroups which corresponds to a subgroup fairness decrease.

This is mostly in line with previous research. Bono et al. (2021) observed that different model types have comparable subgroup performance variation, whereas Gardner et al. (2022) found that subgroup performance variation is lower for random forests than for other models. In our case, overall, random forests showed average performance.

For follow-up work, we see two areas. On the one hand, we suggest to clarify some of the findings of this study. It could be valuable to more closely investigate the causal relationship between the subgroup data characteristics and different models mentioned in Section 3. This could be done by using a regression model with the performance values as target variable and different dataset characteristics and the model types as predictors. One such model could be built per distribution shift and then be used to compare the model coefficients. Conducting this analysis on multiple datasets, for instance on those proposed by Fabris et al. (2022), would further clarify the discussion points in Section 6. On the other hand, the study could be extended to different types of sampling processes for the distribution shifts as well as continuous variables and multiclass problems.

A Appendix

A.1 Literature Research Method

In the course of this work, several literature searches were conducted that are documented in the following two subsections to show how we arrived at the conclusion that related work is rare. This research approach was based on the suggestions of Xiao & Watson (2019) on how to conduct a systematic literature review.

A.1.1 Relationship between Model Choice and Subgroup Fairness

- **Purpose of Study:** Identify previous studies on the performance of subgroups of common classifiers. The main interest is to (A) find out how they compare with each other in terms of subgroup fairness and (B) find theoretical reasons for their behaviour.
- **Research Question(s):** Which classifier is fairer in terms of subgroup fairness?
- **Inclusion Criteria:**
  1. Paper written in English.
  2. Listed on the first five pages of Google Scholar per search expression or referenced in or itself referencing a paper on the topic.
  3. Mentions the empirical or theoretical study of classifiers with respect to subgroup performance in the title or abstract.
- **Search Strategies:**
  - Channels: Google Scholar
  - Keywords:
    - machine learning classifier (Category)
    - subgroup performance, subgroup fairness (Study Subject)
    - study, benchmark, comparison, reasons, understanding, achieving (Investigation Method)
  - Search expressions: (created based on a combination of keywords)
    - Machine Learning Classifiers Subgroup Performance Benchmark
    - Comparison of Machine Learning Classifiers with respect to Subgroup Performance
    - Understanding differing Subgroup Performance of Machine Learning Classifiers
    - Reasons for differing Subgroup Fairness of Machine Learning Classifiers
    - Study of Subgroup Fairness among Machine Learning Classifiers
    - Achieve Subgroup Fairness through Model Choice

        - Role of Machine Learning Model Choice on Subgroup Accuracy
        - Forward/Backward Search from Popular Papers in Subgroup Fairness:
            - ”Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification” by Joy Buolamwini and Timnit Gebru (Buolamwini & Gebru, 2018)
            - ”Multiaccuracy: Black-Box Post-Processing for Fairness in Classification” by Michael P. Kim et al. (Kim, Ghorbani, & Zou, 2018)
            - ”Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness” by Michael Kearns et al. (Kearns et al., 2018)
            - ”An Empirical Study of Rich Subgroup Fairness for Machine Learning” by Michael Kearns et al. (Kearns et al., 2019)
            - Multicalibration: Calibration for the (computationally-identifiable) masses (Hebert-Johnson et al., 2018)

A.1.2 Relationship between Model Choice and Subgroup Fairness under Distribution Shift

- **Purpose of Study:** Identify previous studies on the impact of distribution shifts on subgroup fairness of machine learning classifiers. The main interest is to (A) find out how they compare against each other (B) find theoretical reasons for their behaviour.

- **Research Question(s):** What impact do distribution shifts have on the subgroup fairness of machine learning classifiers?

- **Inclusion Criteria:**

    1. Paper written in English.
    2. Listed on the first five pages of Google Scholar per search expression or referenced in or itself referencing a paper on the topic.
    3. Mentions the empirical or theoretical study of classifiers with respect to distribution shifts in the title or abstract.

- **Search Strategies:**

    - Channels: Google Scholar
    - Keywords:
        - machine learning classifier, classifier names (Category)
        - behaviour under/robustness towards/influence/effect of distribution shifts/data set shifts on subgroup performance (Study Subject)
        - study, understanding, reasons, investigation, benchmark, evaluation, comparison (Investigation Method)

- Search expressions: (created based on a combination of keywords)
  - ∗ Influence of Distribution Shifts on Subgroup Performance of Machine Learning Classifiers
  - ∗ Influence of Data Set Shifts on Subgroup Performance of Machine Learning Classifiers
  - ∗ Benchmark Machine Learning Classifiers Distribution Shift
  - ∗ Effect of Distribution Shifts on Random Forest
  - ∗ Evaluation of Support Vector Machines in the case of Data Set Shifts
  - ∗ Robustness of Logistic Regression against Distribution Shifts
  - ∗ Robustness against Shifts in Sociodemographic Groups Machine Learning

A.2 Descriptive Analysis

A.2.1 Descriptive Analysis of Adult Dataset

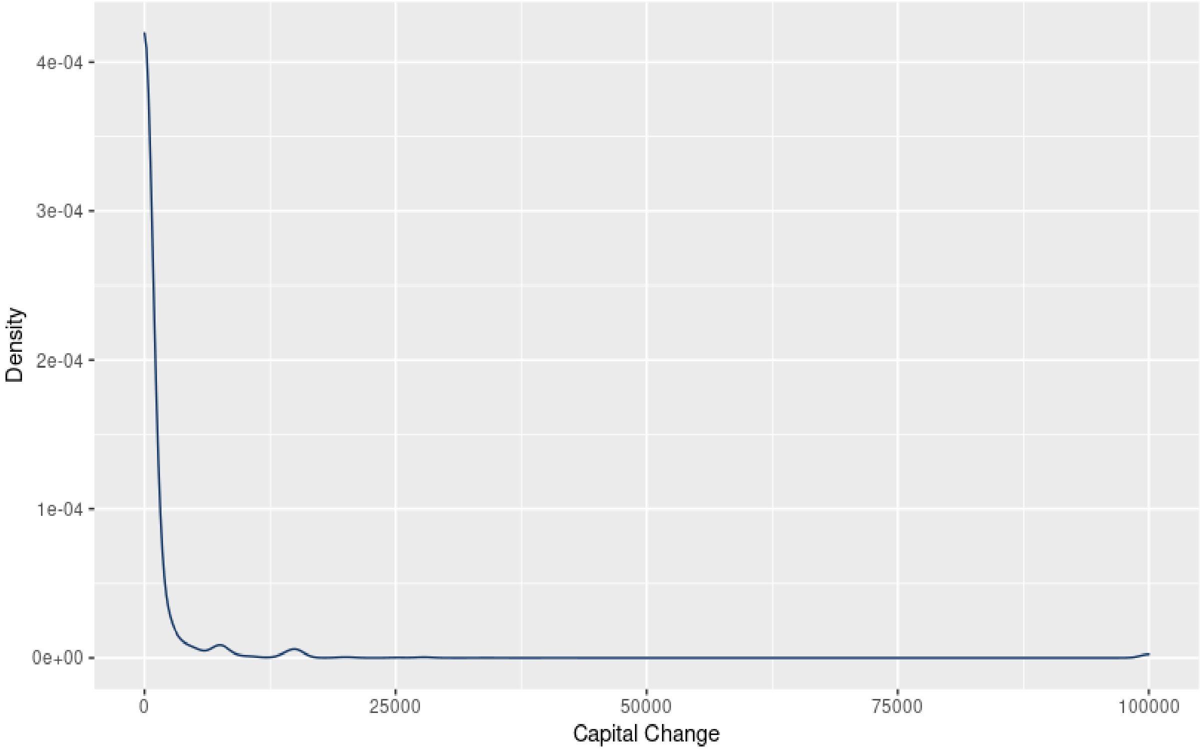

Figure 11: Distribution of the variable 'Capital Change' in the dataset

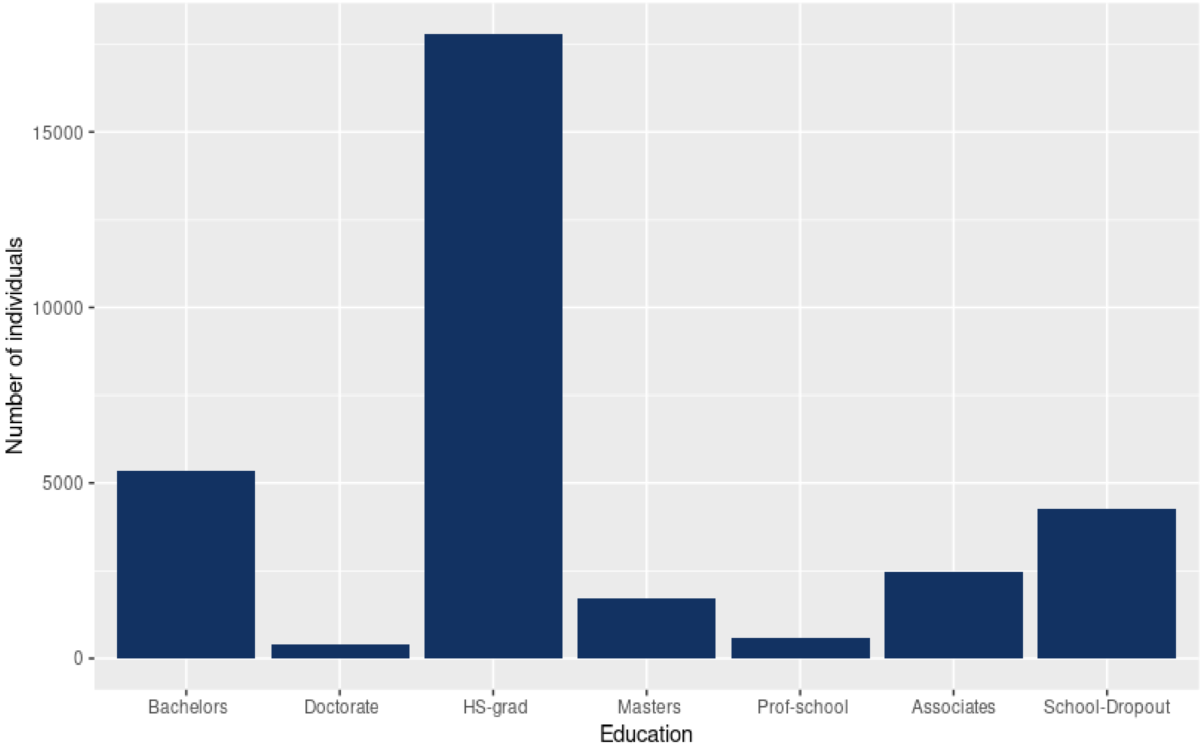

Figure 12: Distribution of the variable 'Education' in the dataset

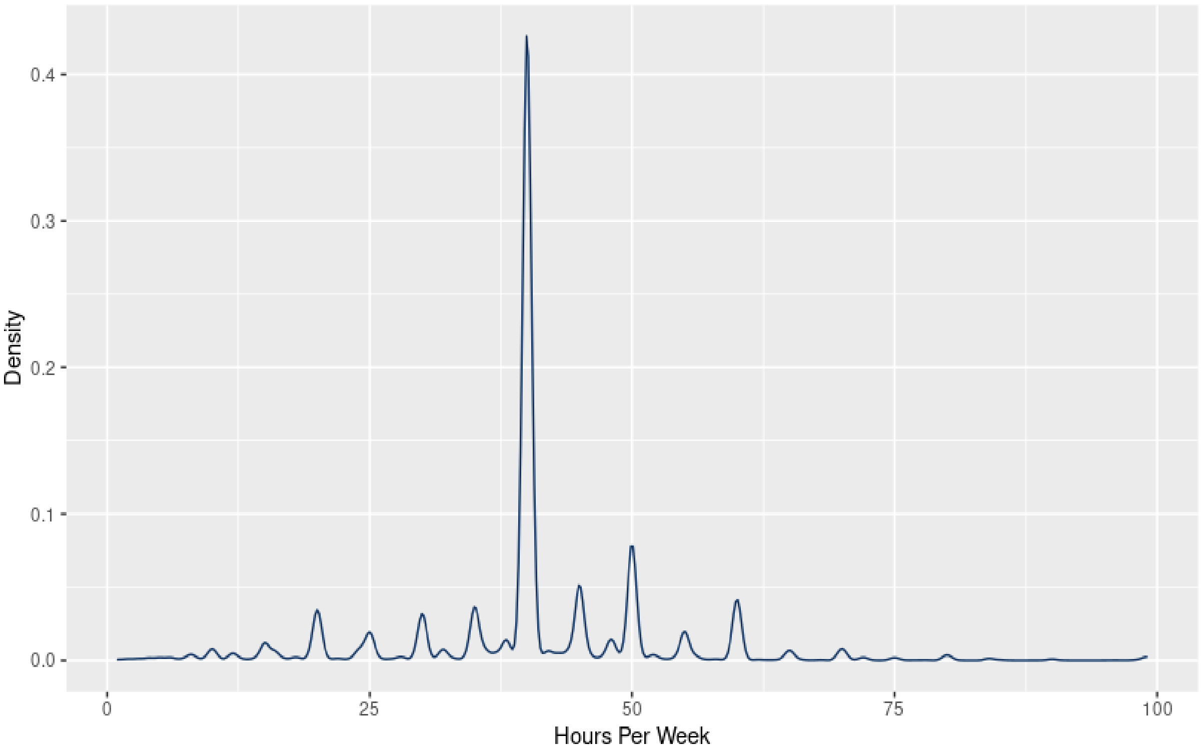

Figure 13: Distribution of the variable 'Hours per Week' in the dataset

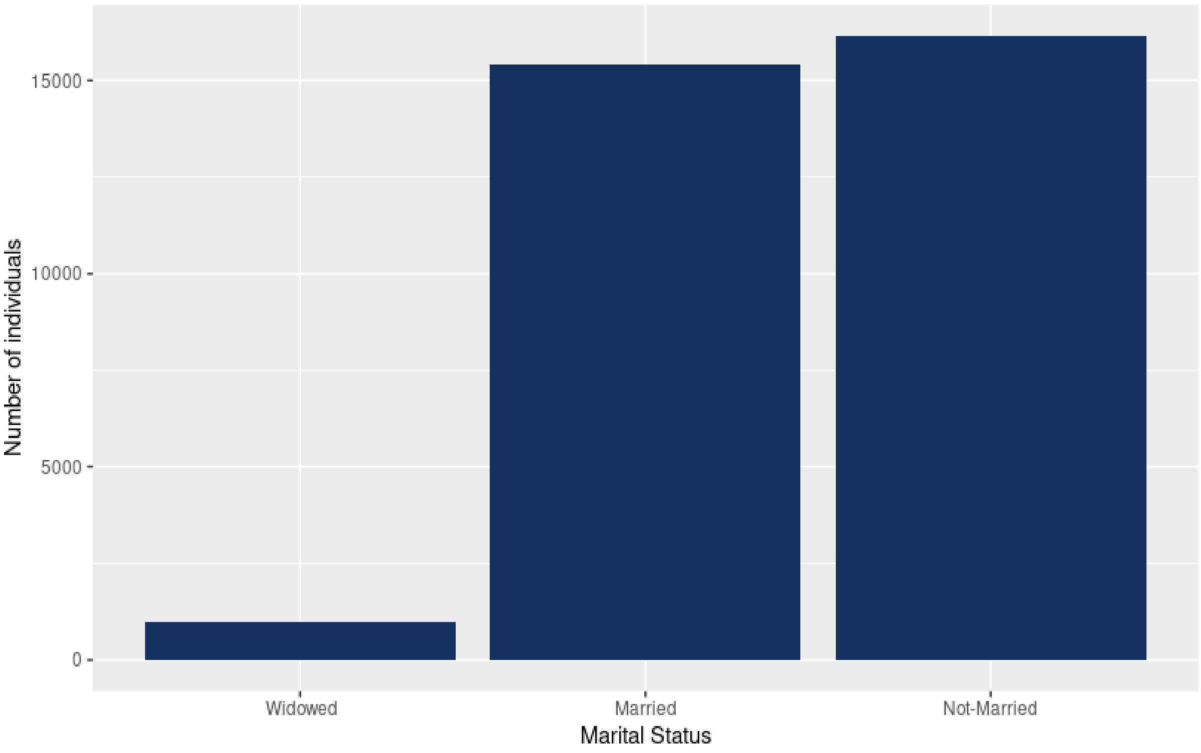

Figure 14: Distribution of the variable 'Marital Status' in the dataset

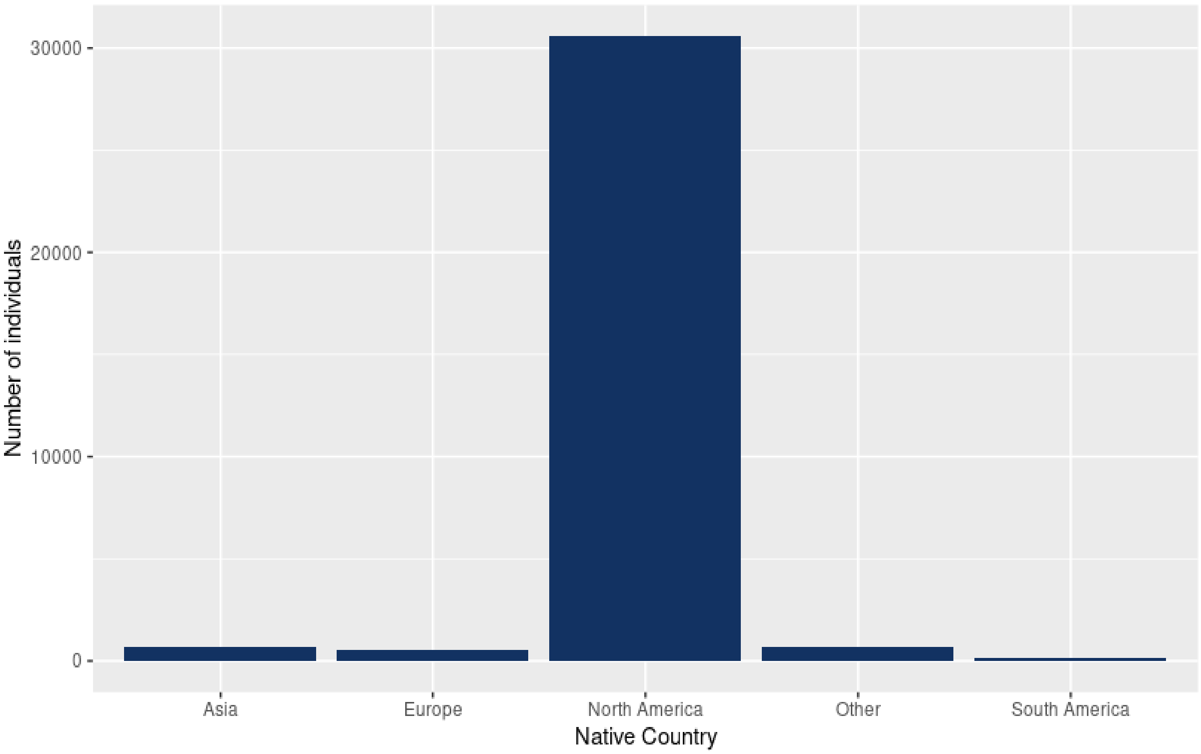

Figure 15: Distribution of the variable ’Native Country’ in the dataset

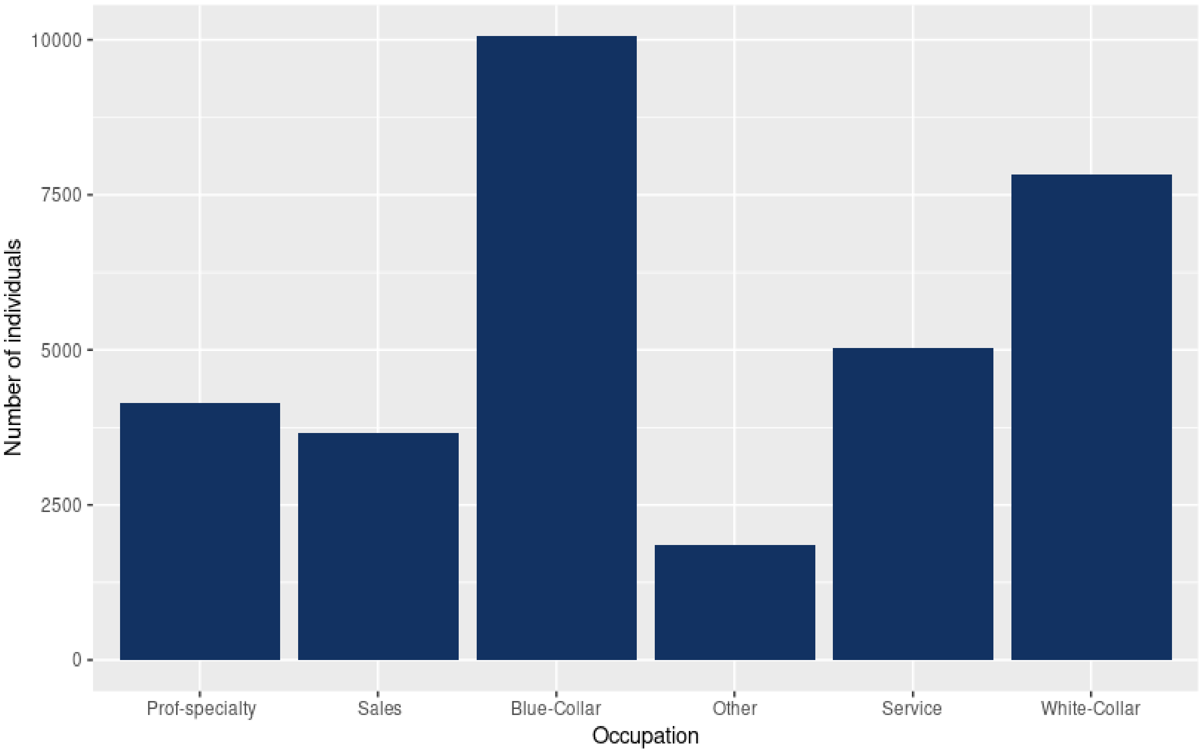

Figure 16: Distribution of the variable ’Occupation’ in the dataset

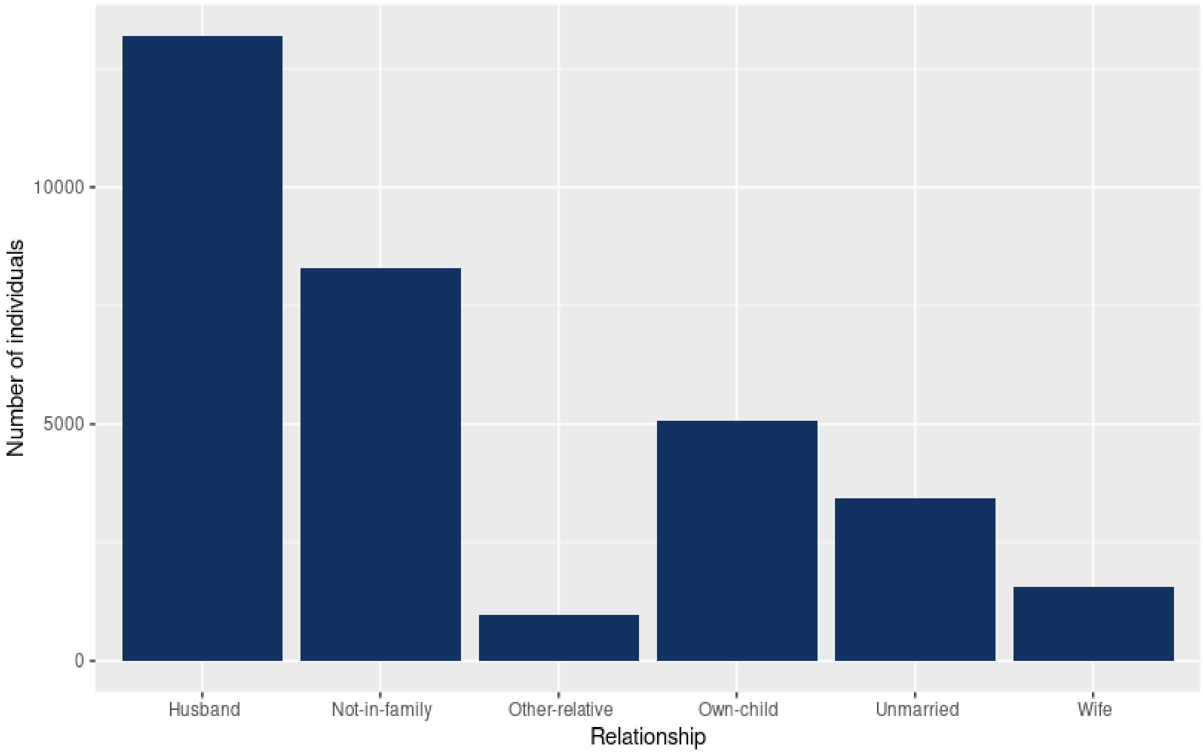

Figure 17: Distribution of the variable 'Relationship' in the dataset

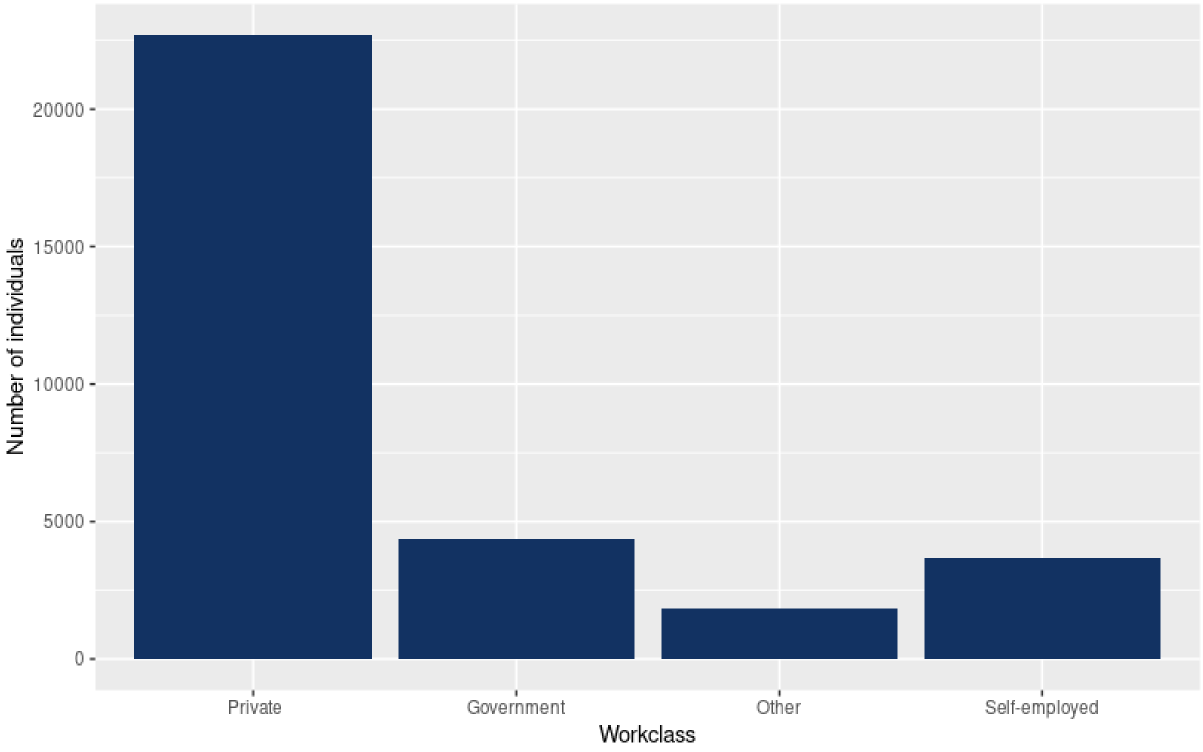

Figure 18: Distribution of the variable 'Workclass' in the dataset

A.2.2 Descriptive Analysis of Subgroups

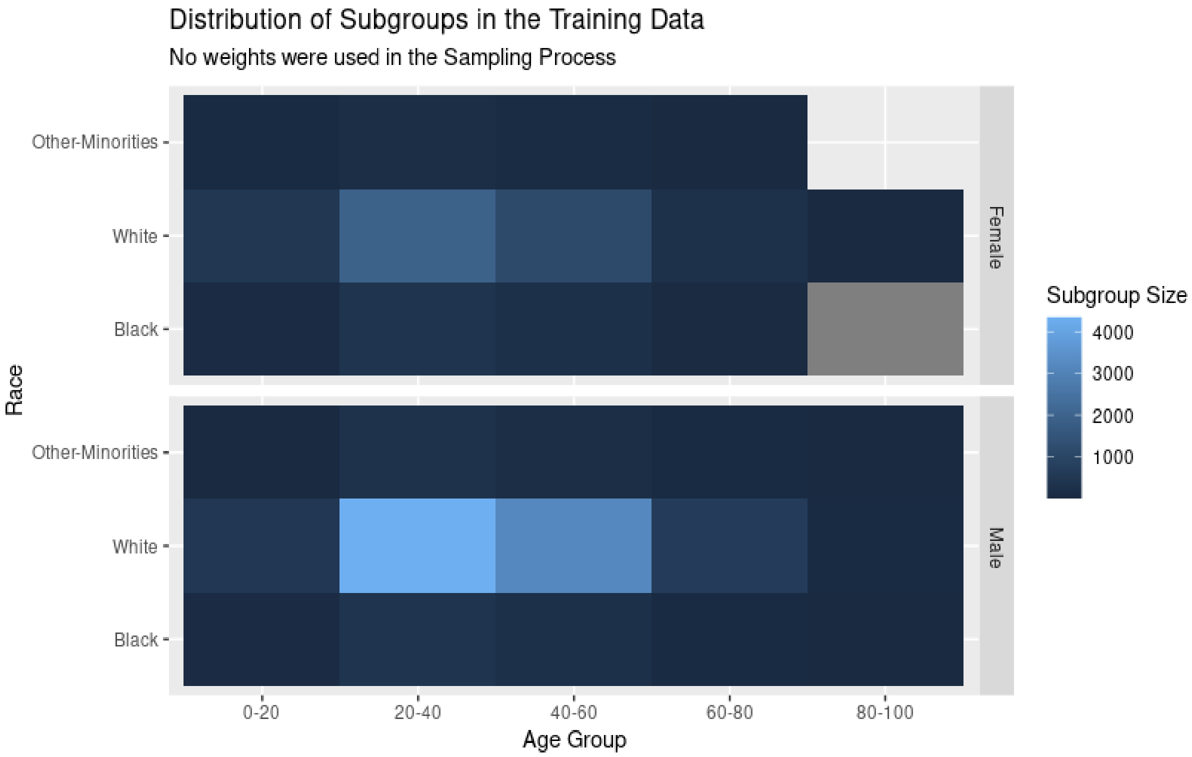

Figure 19: Group Size per Subgroup (Training Data, *No Weights*)

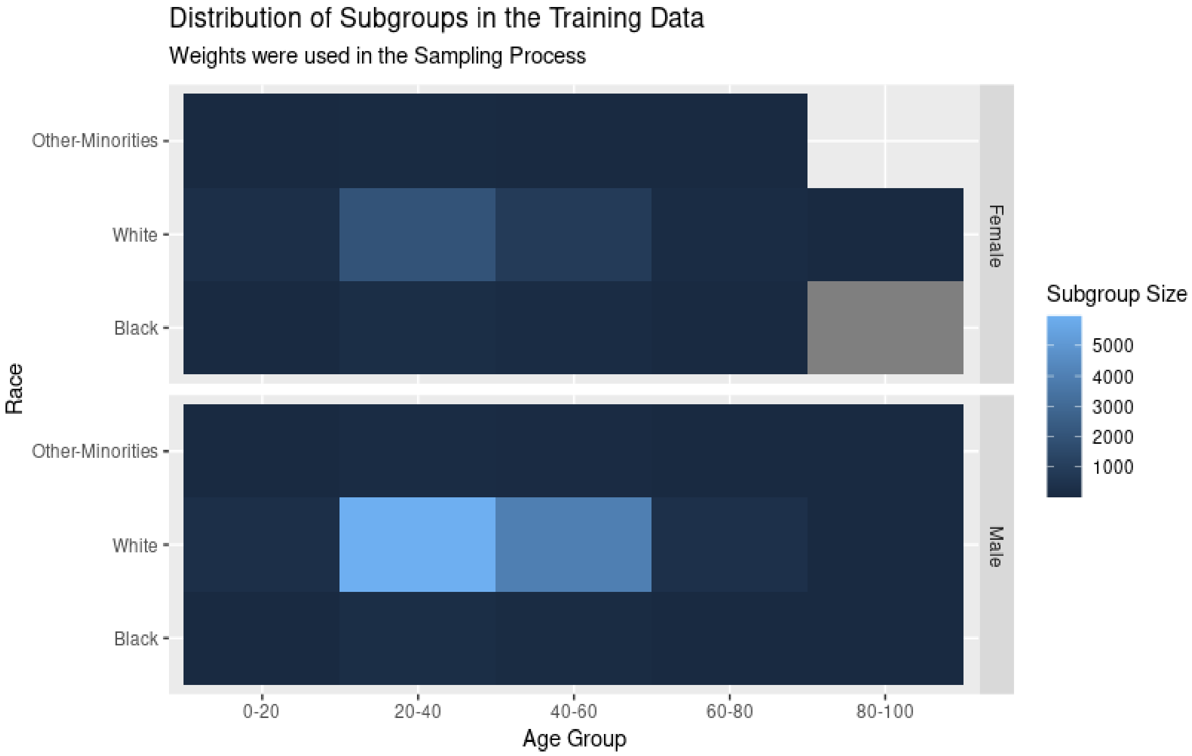

Figure 20: Group Size per Subgroup (Training Data, *Weights*)

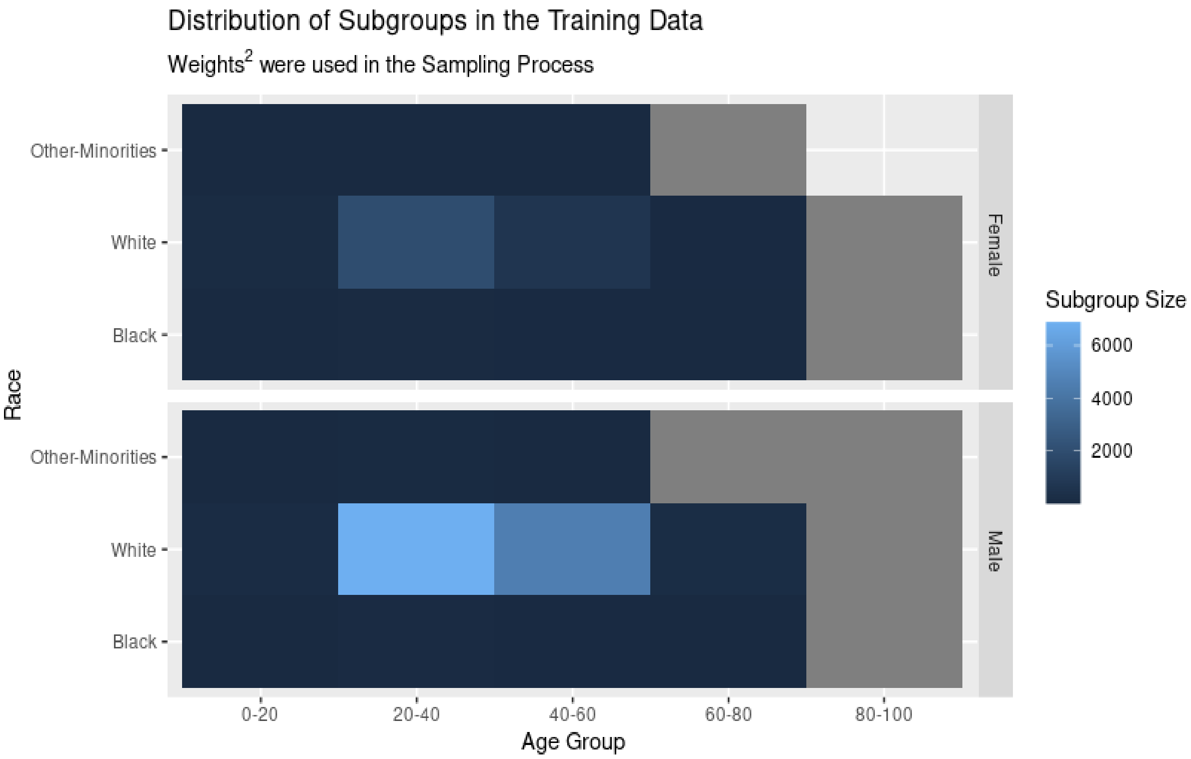

Figure 21: Group Size per Subgroup (Training Data, $Weights^2$)

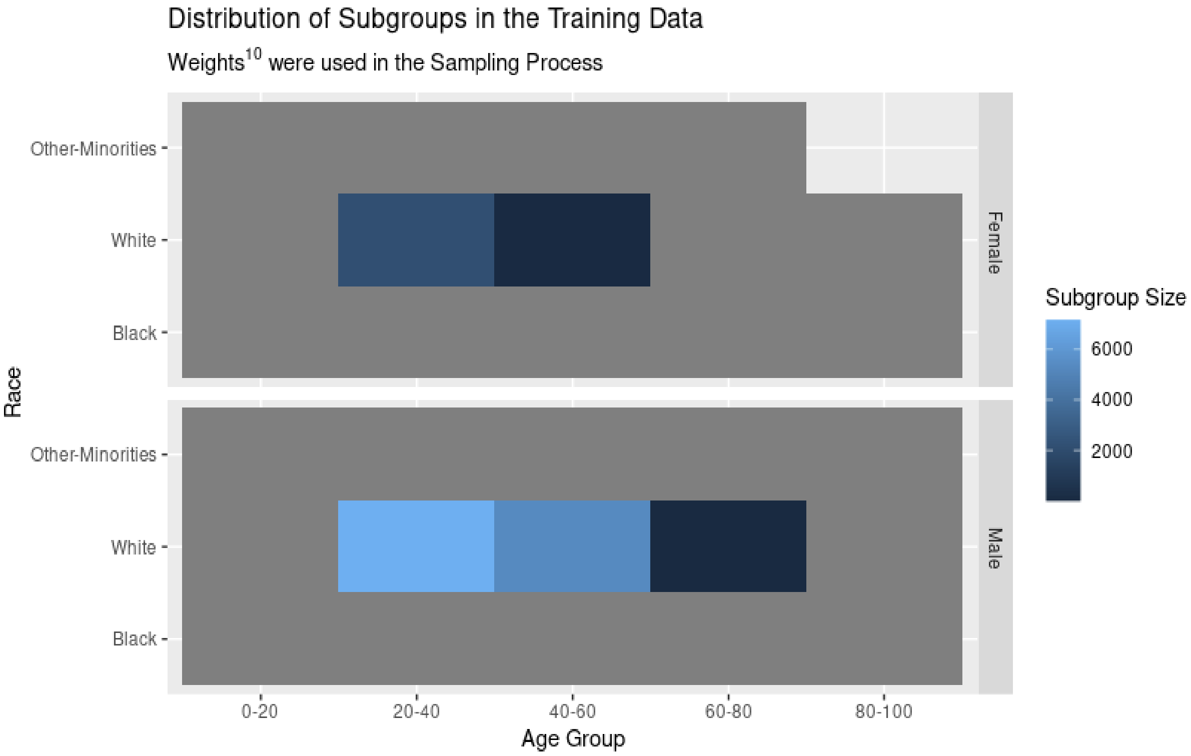

Figure 22: Group Size per Subgroup (Training Data, $Weights^{10}$)

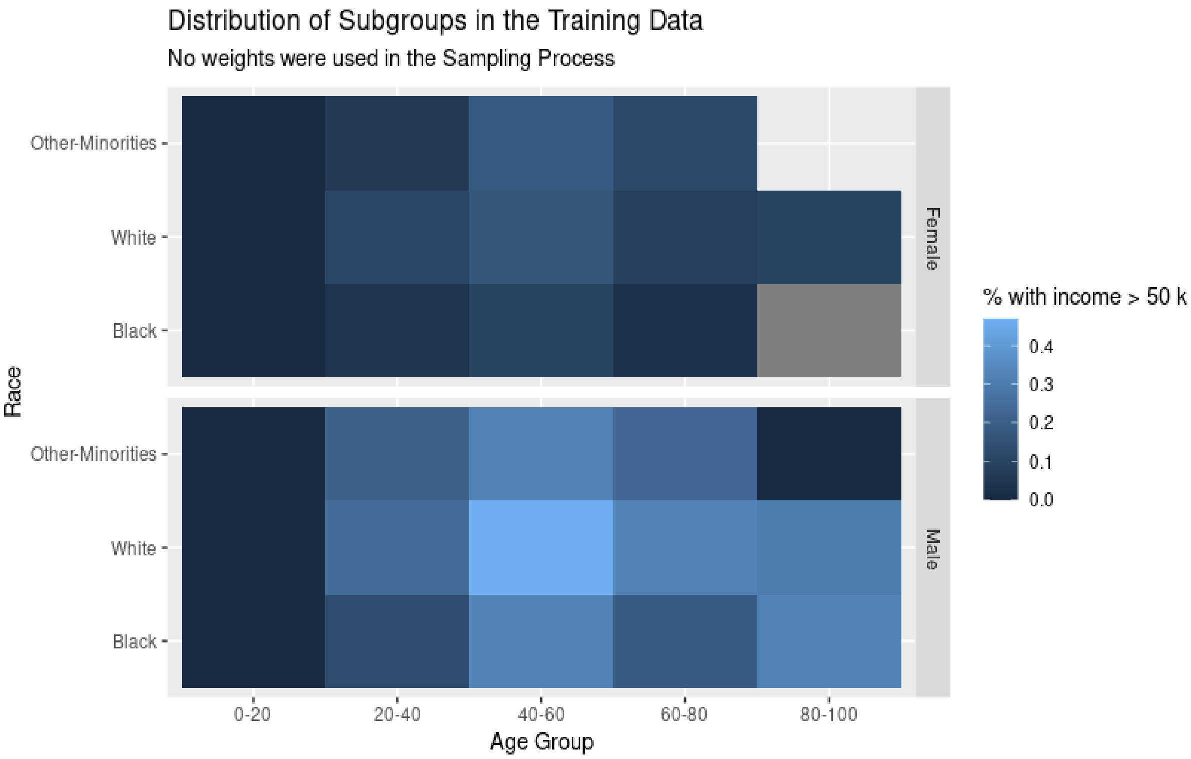

Figure 23: Income Distribution per Subgroup (Training Data, $No\ Weights$)

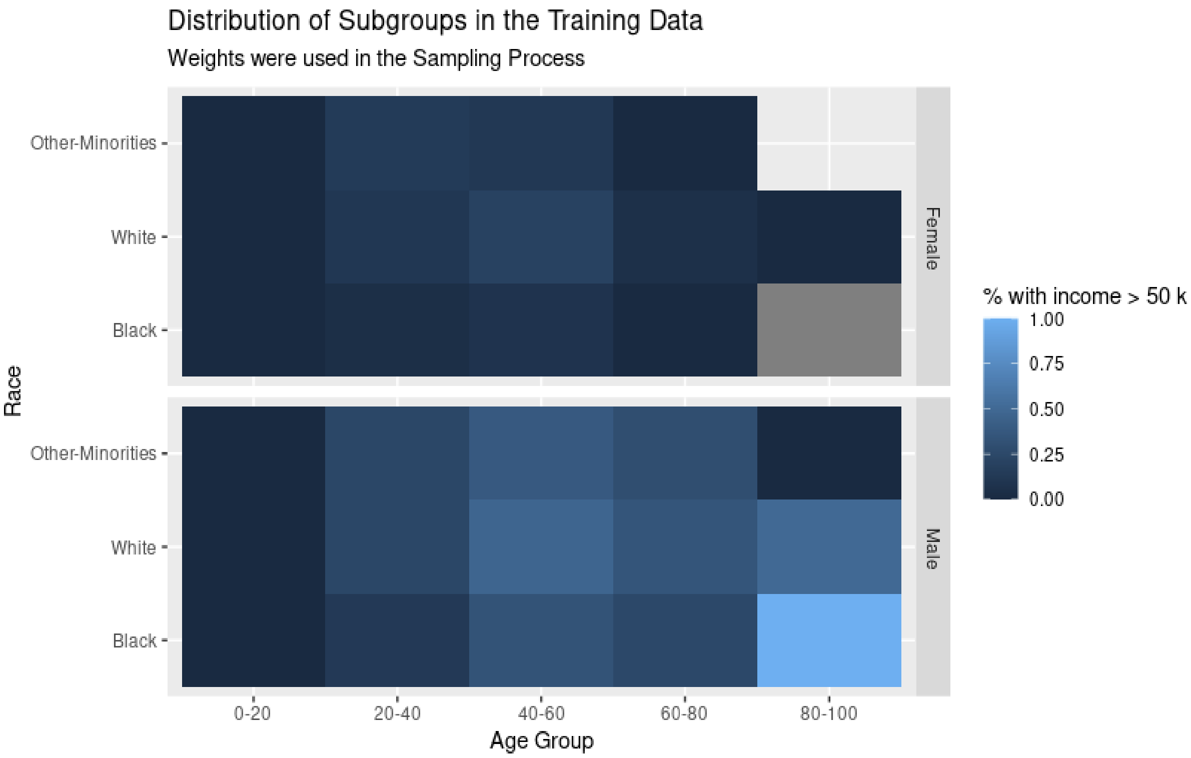

Figure 24: (Training Data, $Weights$)

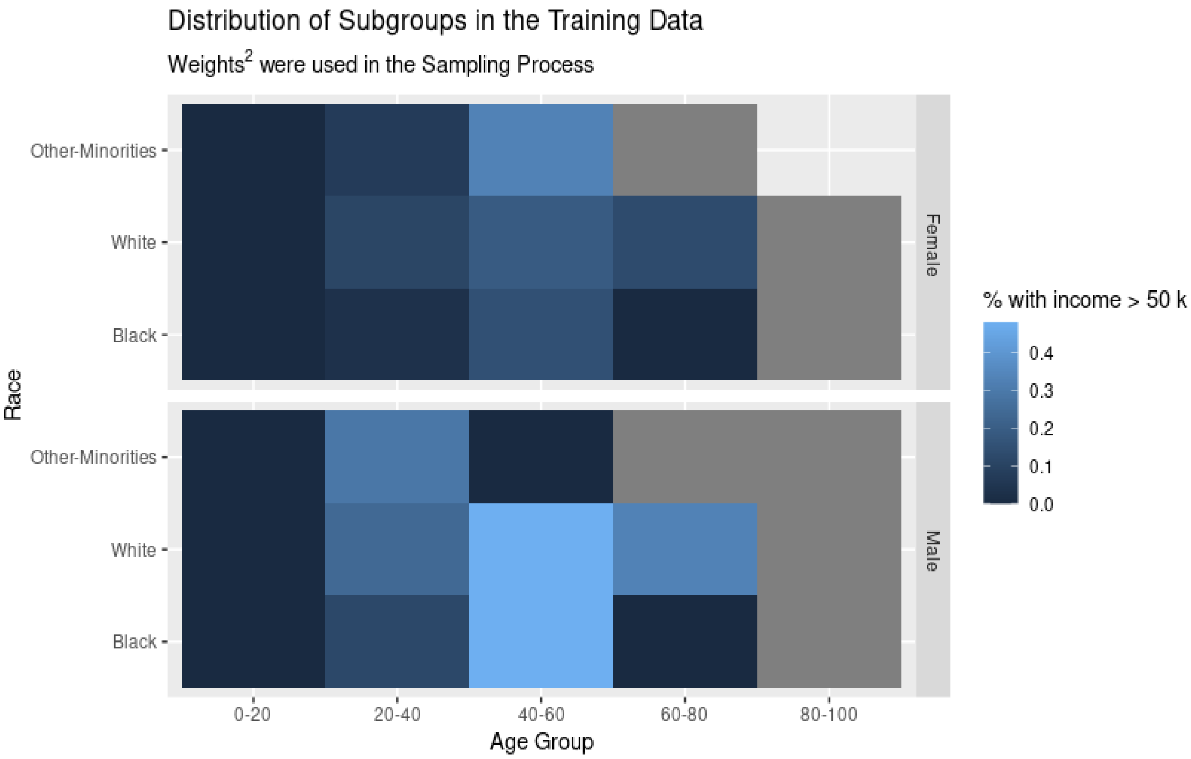

Figure 25: Income Distribution per Subgroup (Training Data, $Weights^2$)

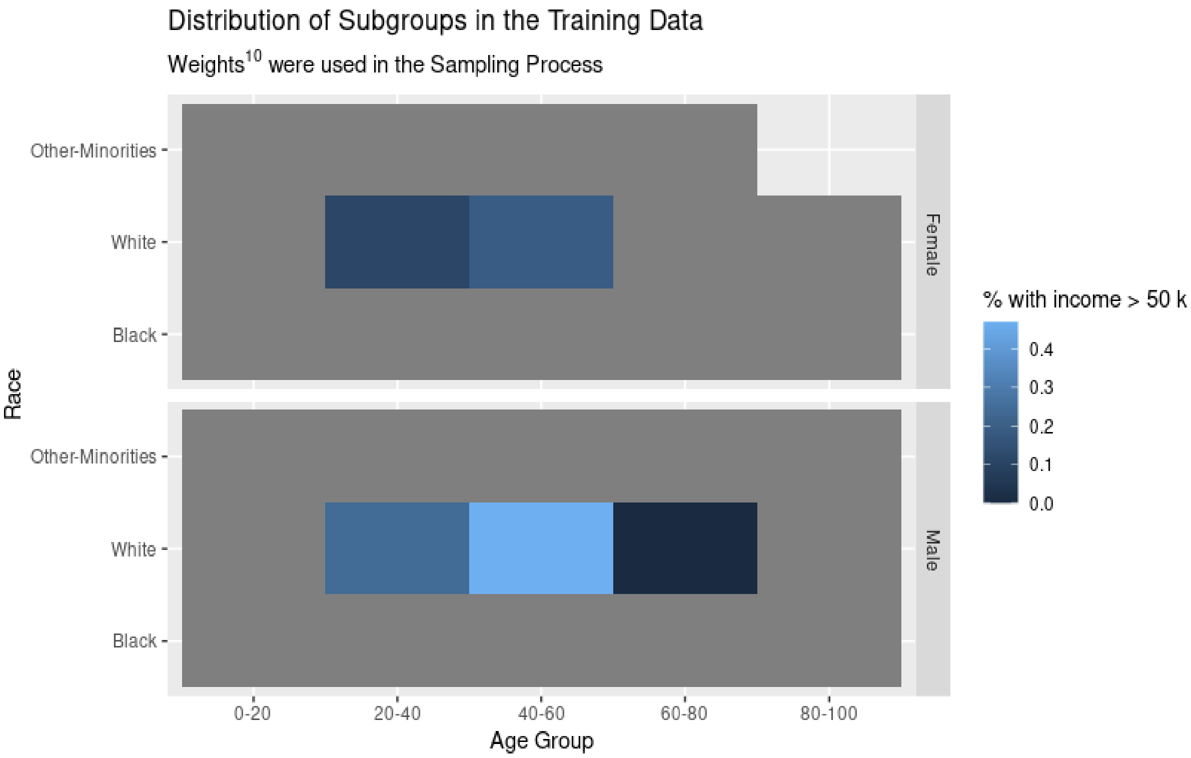

Figure 26: Income Distribution per Subgroup (Training Data, $Weights^{10}$)

A.3 Predictors per Training Sample

Model Predictors				
Variable	**Unshifted**	**Weights^1**	**Weights^2**	**Weights^{10}**
Education (Associates)	x	x	x	x
Education (HS-grad)	x	x	x	x
Education (Masters)	x	x	x	x
Education (School Dropout)	x	x	x	x
Hours Per Week	x	x	x	x
Marital Status (Not-Married)	x	x	x	x
Native Country (North America)	x	-	-	-
Occupation (Blue-Collar)	x	x	x	x
Occupation (Sales)	x	x	x	x
Occupation (Service)	x	x	x	x
Occupation (White-Collar)	x	x	x	x
Relationship (Not-in-family)	x	x	x	x
Relationship (Own-child)	x	x	x	x
Relationship (Unmarried)	x	x	x	x
Workclass (Government)	x	x	x	x
Workclass (Other)	x	-	-	-
Workclass (Self-employed)	x	x	x	x

Table 7: This table contains all the dummy encoded variables that were used to build the models under the respective shift. Some categories of some categorical variables have been removed due to sparsity or collinearity.

A.4 Subgroup Performance Results (complete)

A.4.1 Subgroup Performance per Classifier

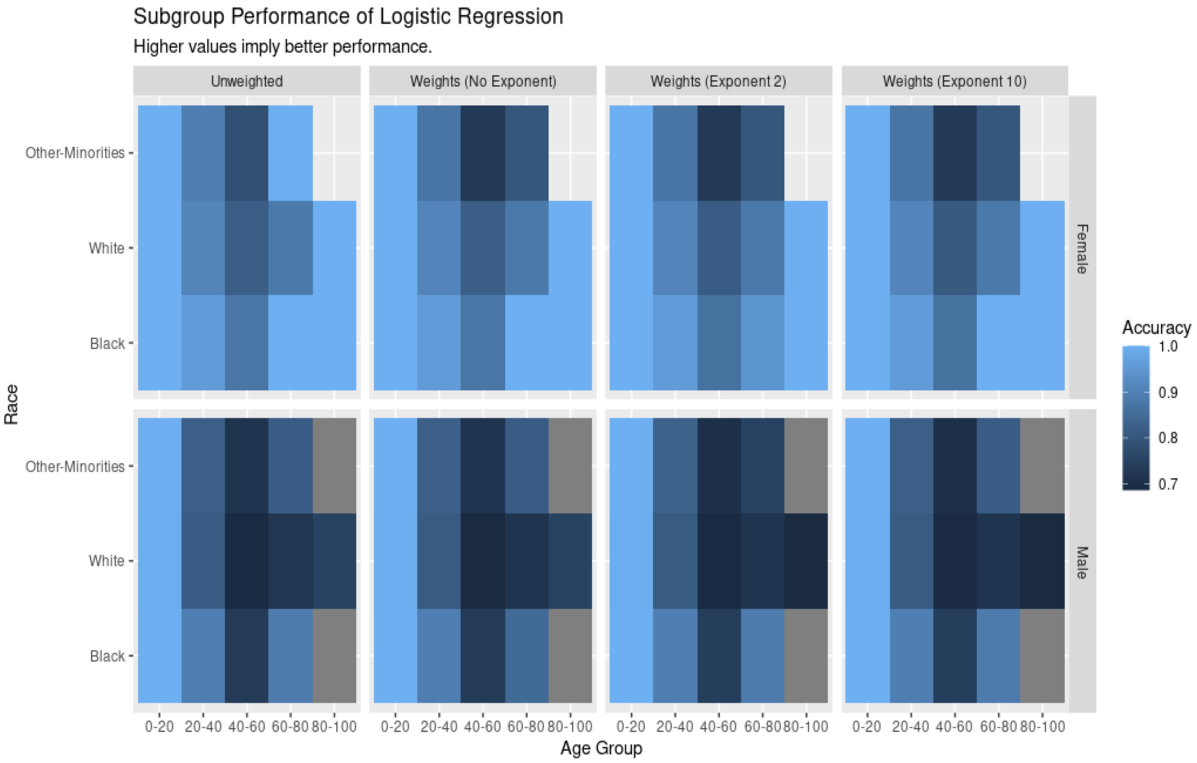

Figure 27: Subgroup Performance of Logistic Regression (Acc)

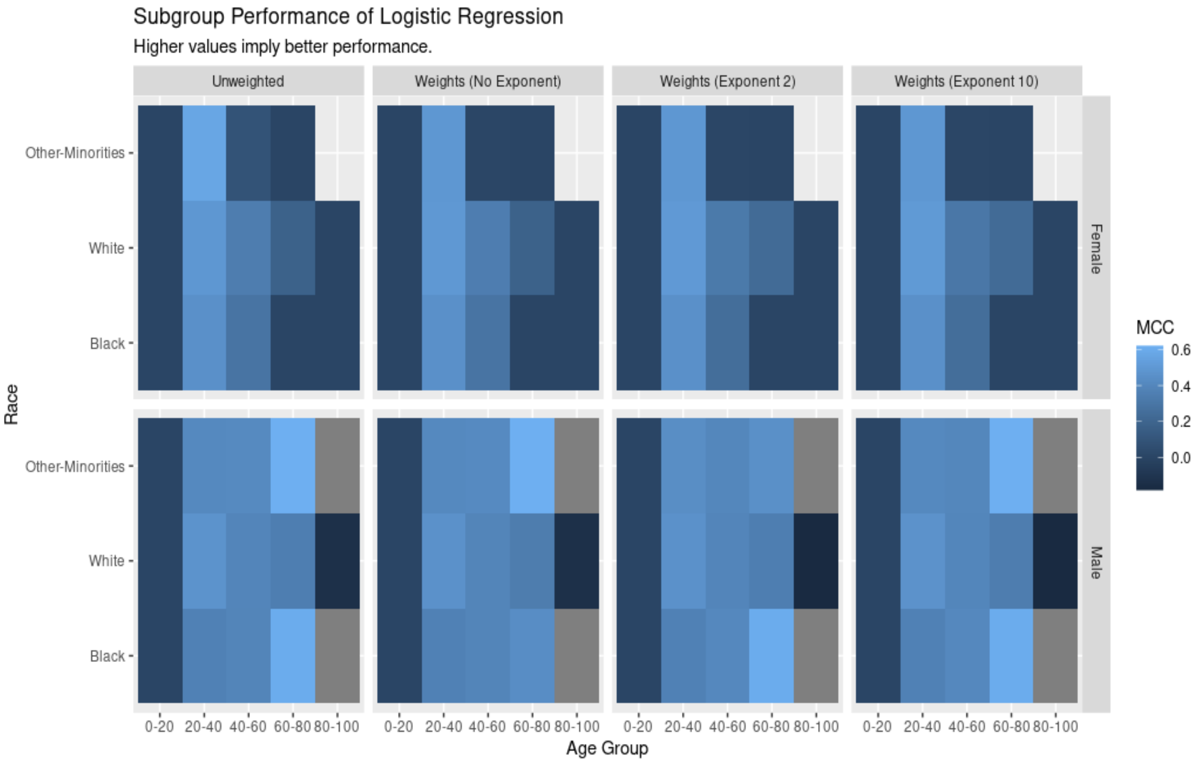

Figure 28: Subgroup Performance of Logistic Regression (MCC)

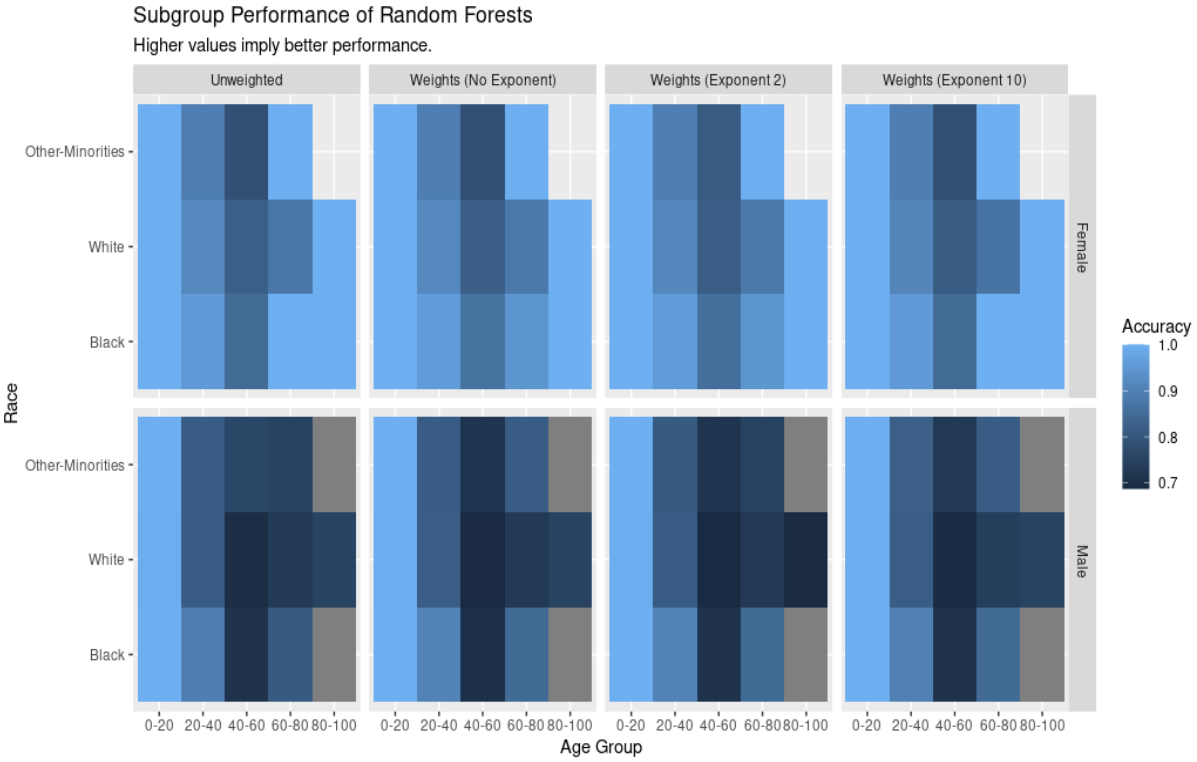

Figure 29: Subgroup Performance of Random Forests (Acc)

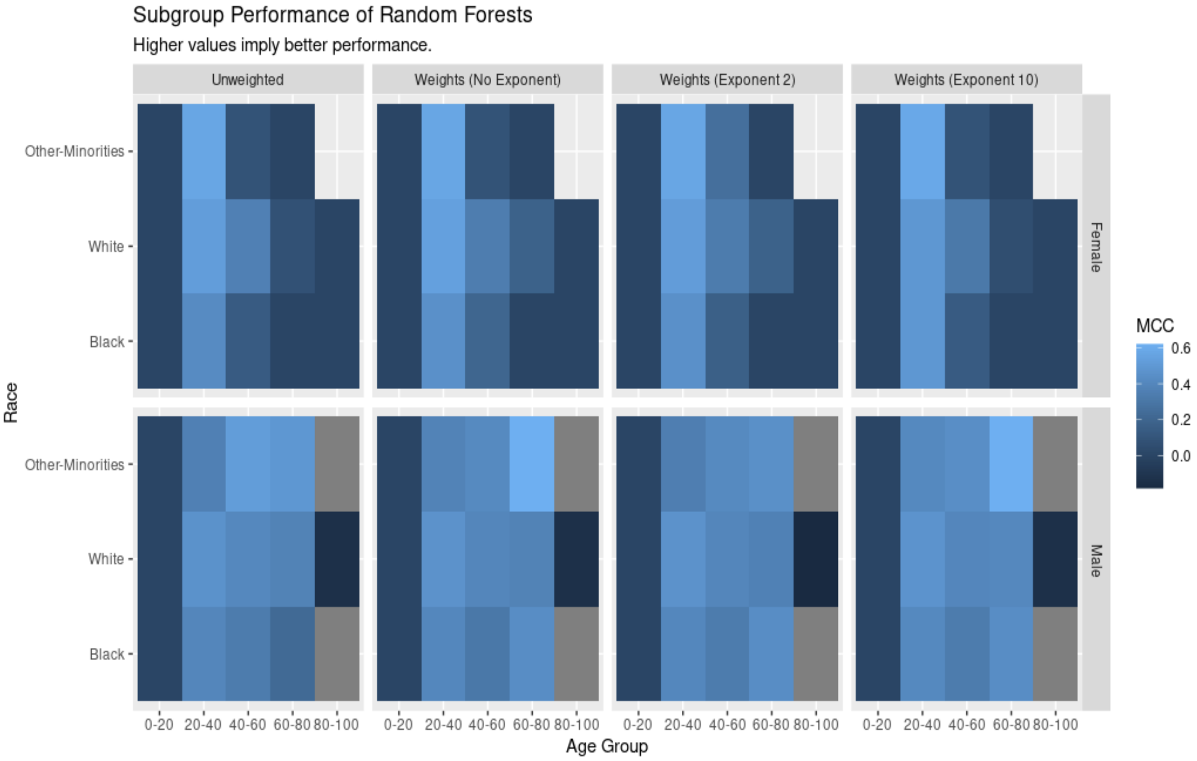

Figure 30: Subgroup Performance of Random Forests (MCC)

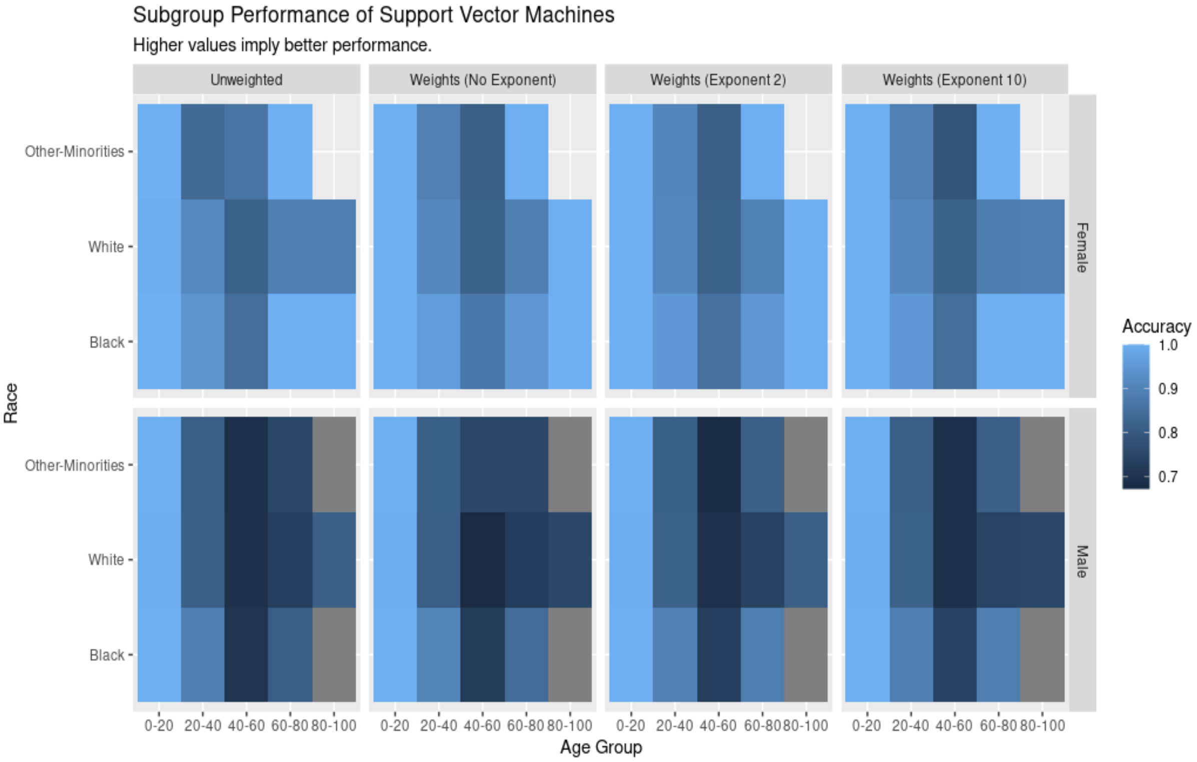

Figure 31: Subgroup Performance of Support Vector Machines (Acc)

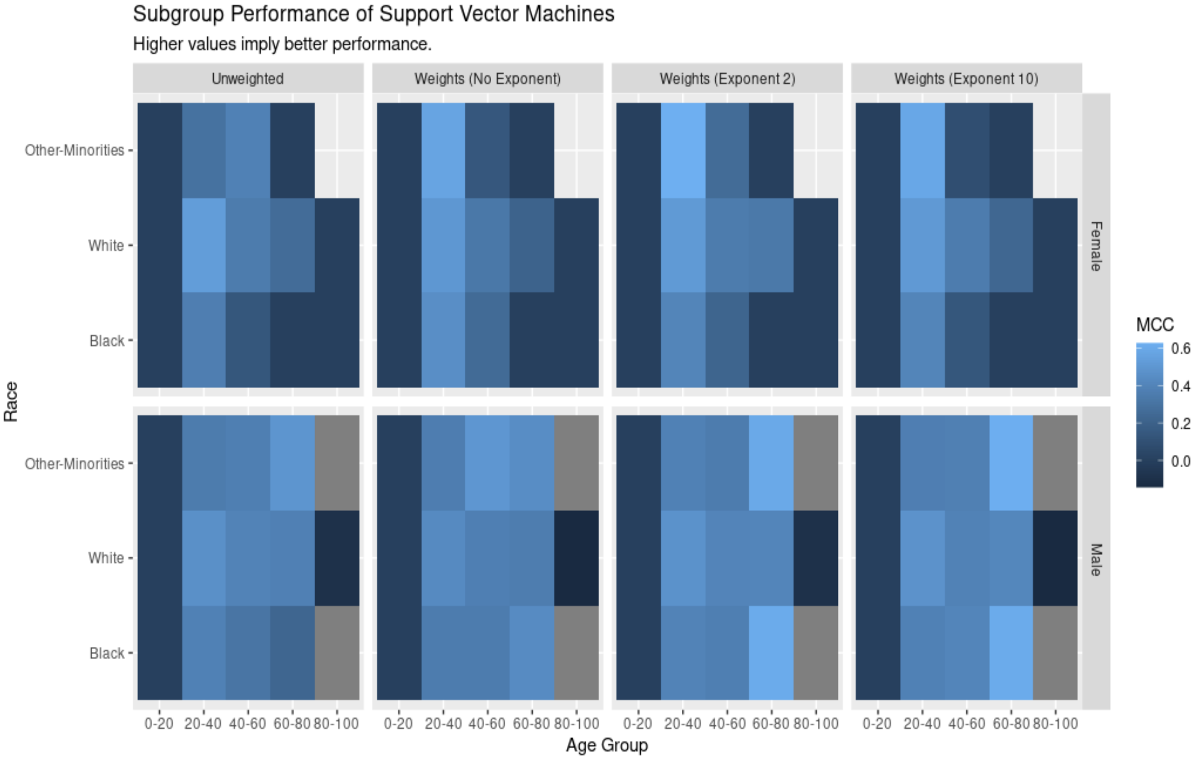

Figure 32: Subgroup Performance of Support Vector Machines (MCC)

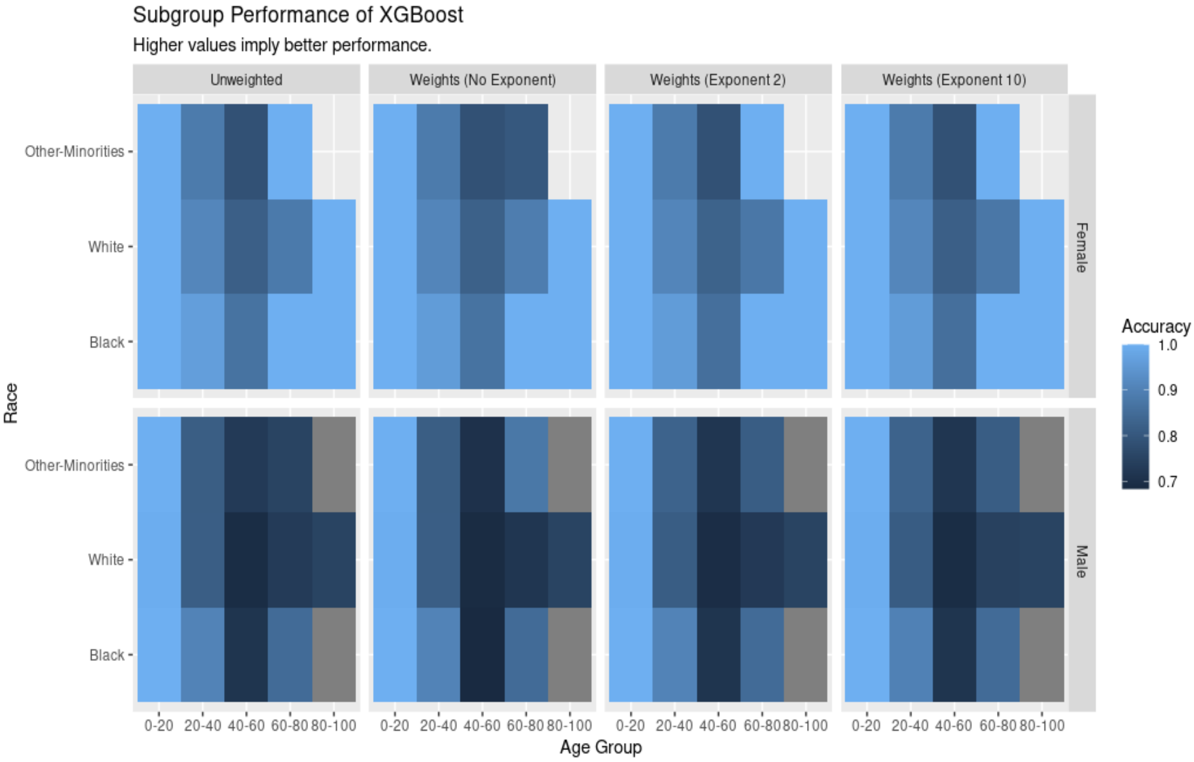

Figure 33: Subgroup Performance of XGBoost (Acc)

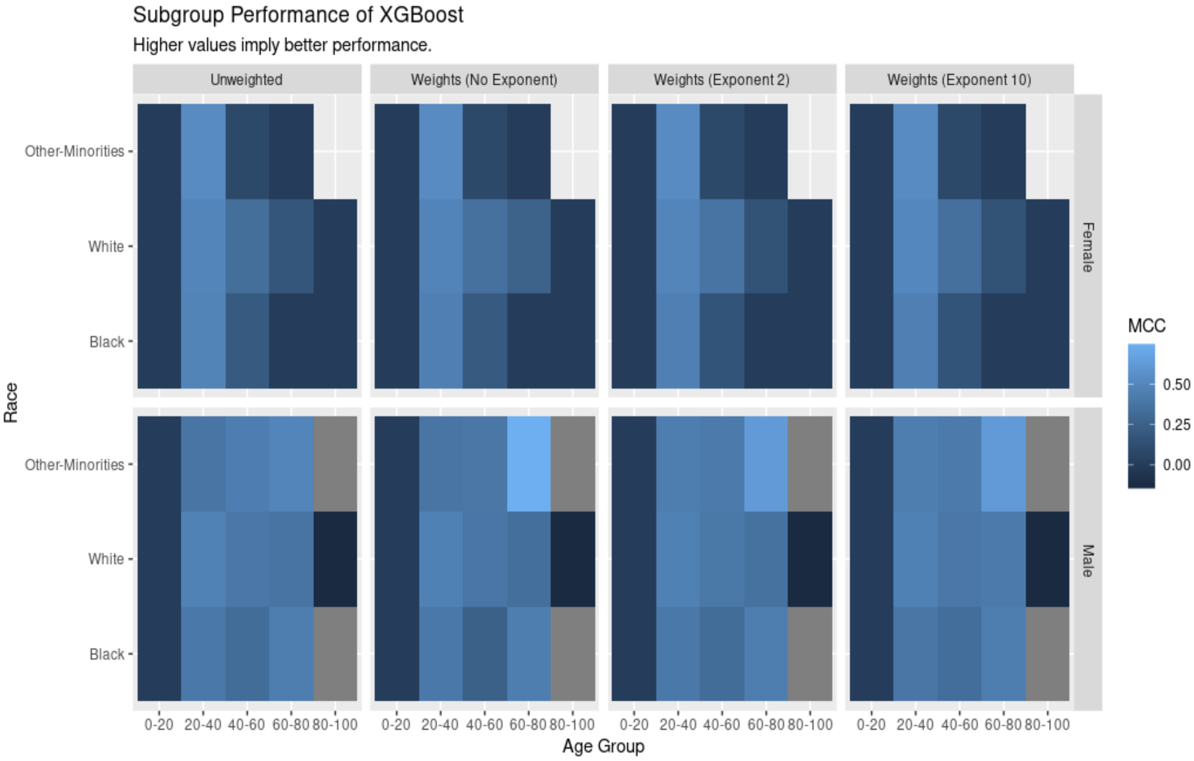

Figure 34: Subgroup Performance of XGBoost (MCC)

A.4.2 Classifier Comparison

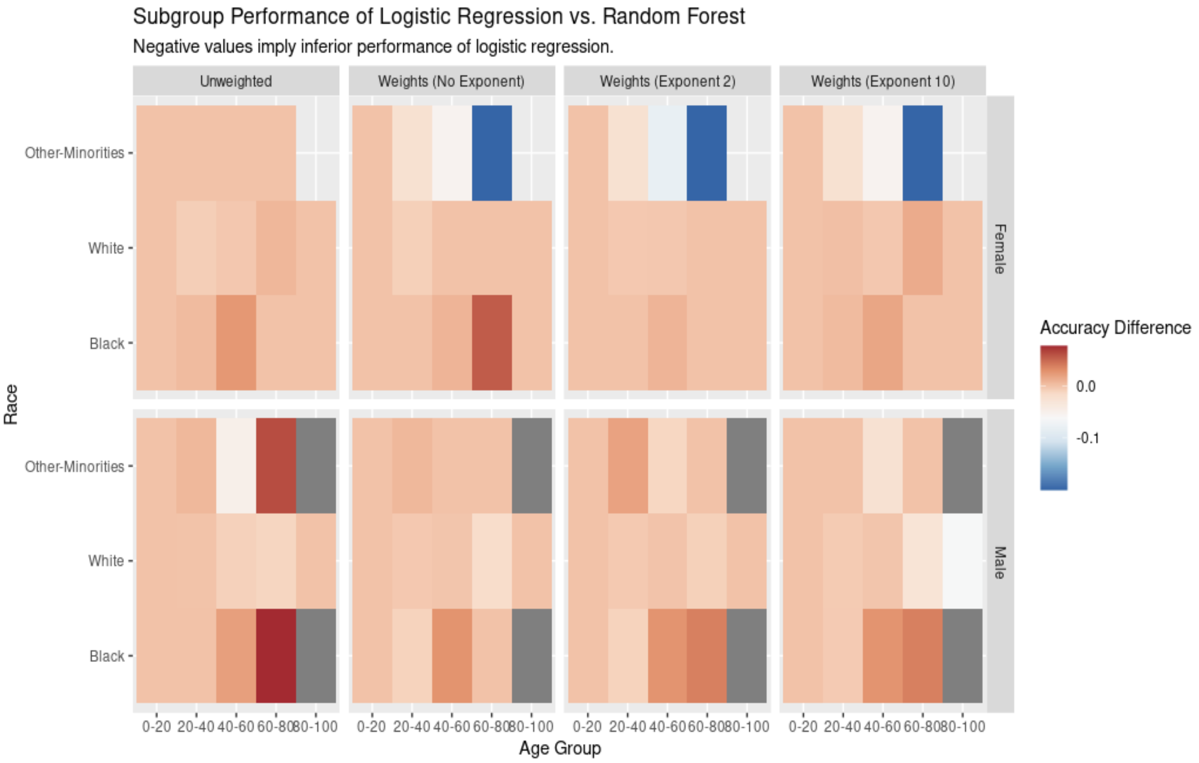

Figure 35: Subgroup Performance Difference between Logistic Regression and Random Forests (Acc)

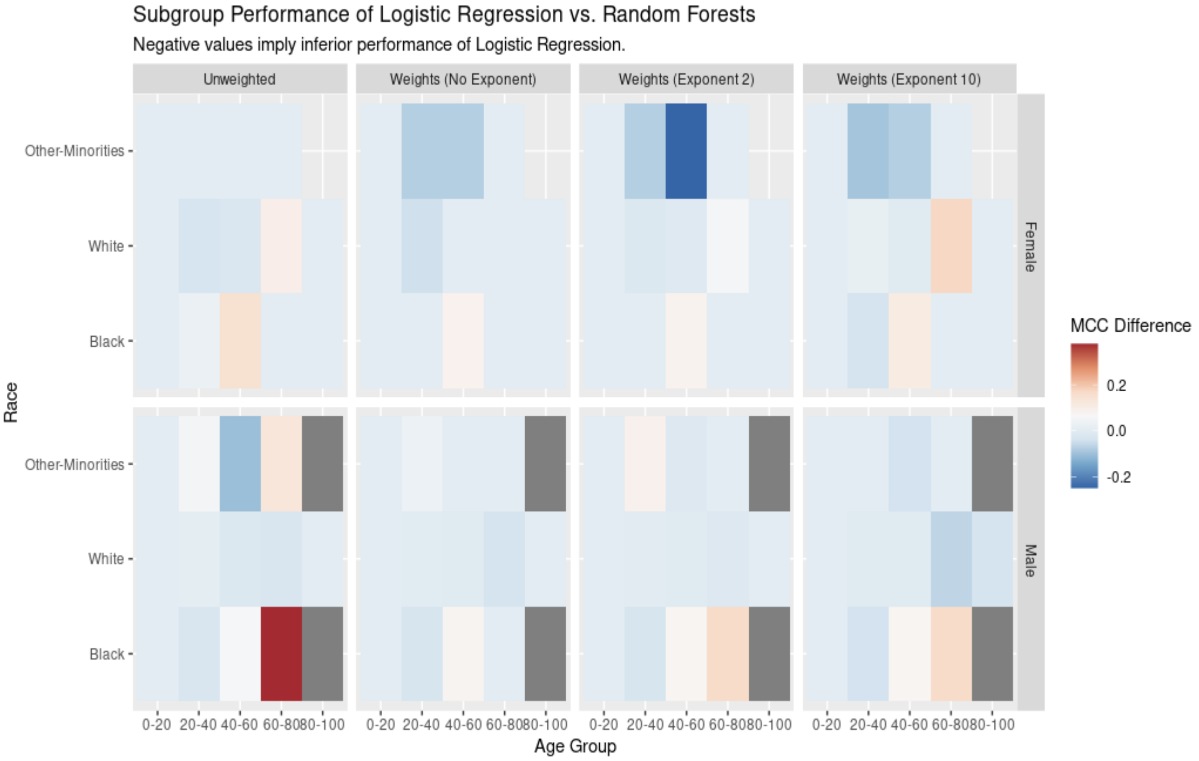

Figure 36: Subgroup Performance Difference between Logistic Regression and Random Forests (MCC)

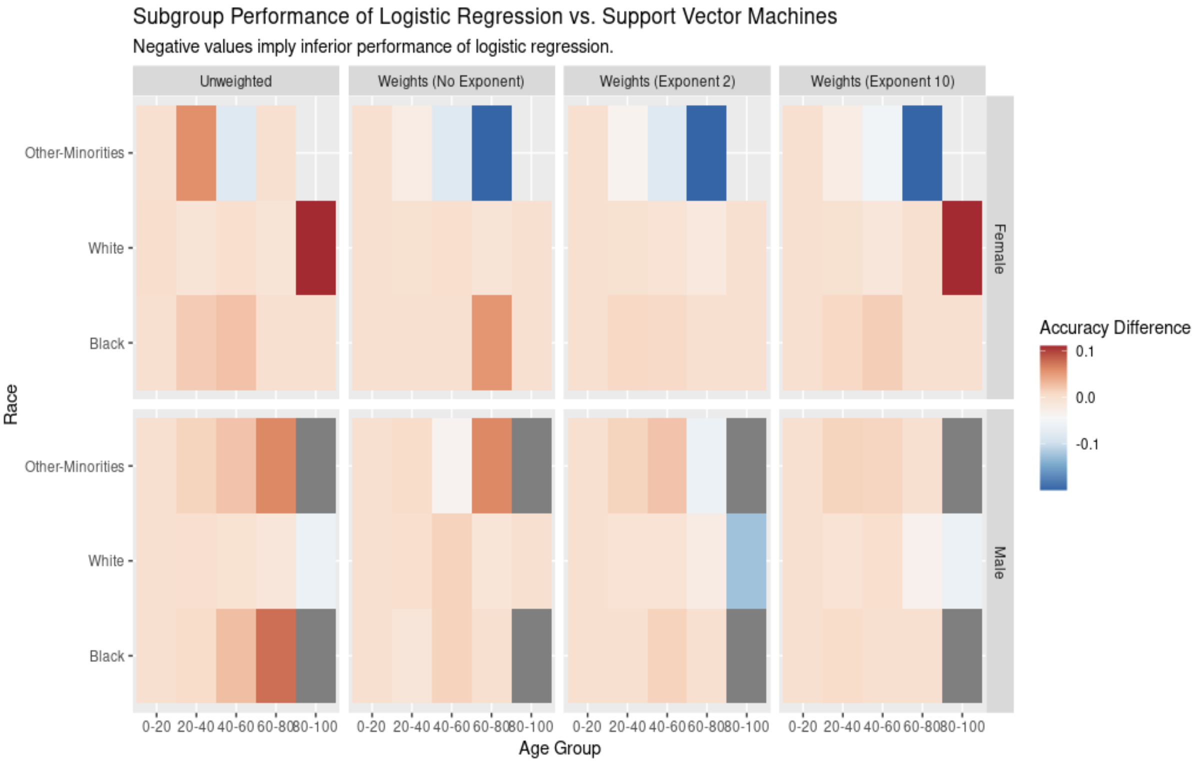

Figure 37: Subgroup Performance Difference between Logistic Regression and Support Vector Machines (Acc)

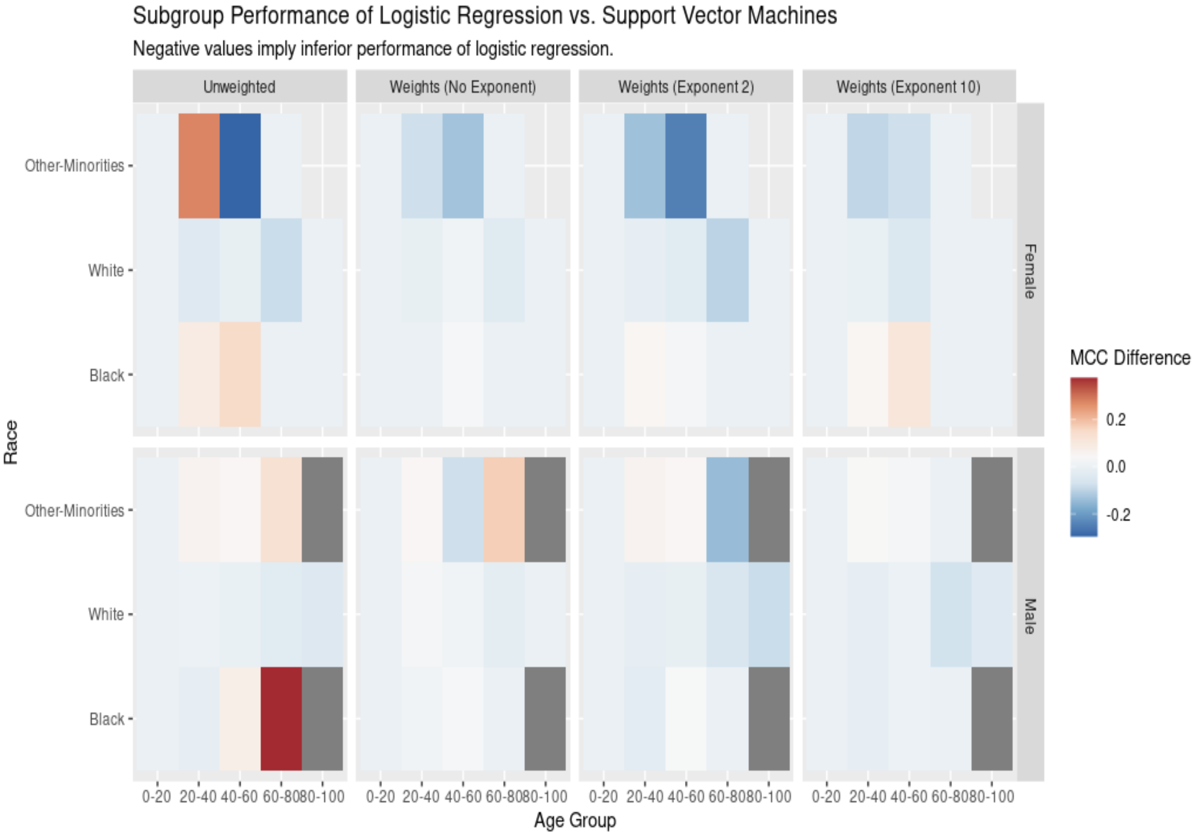

Figure 38: Subgroup Performance Difference between Logistic Regression and Support Vector Machines (MCC)

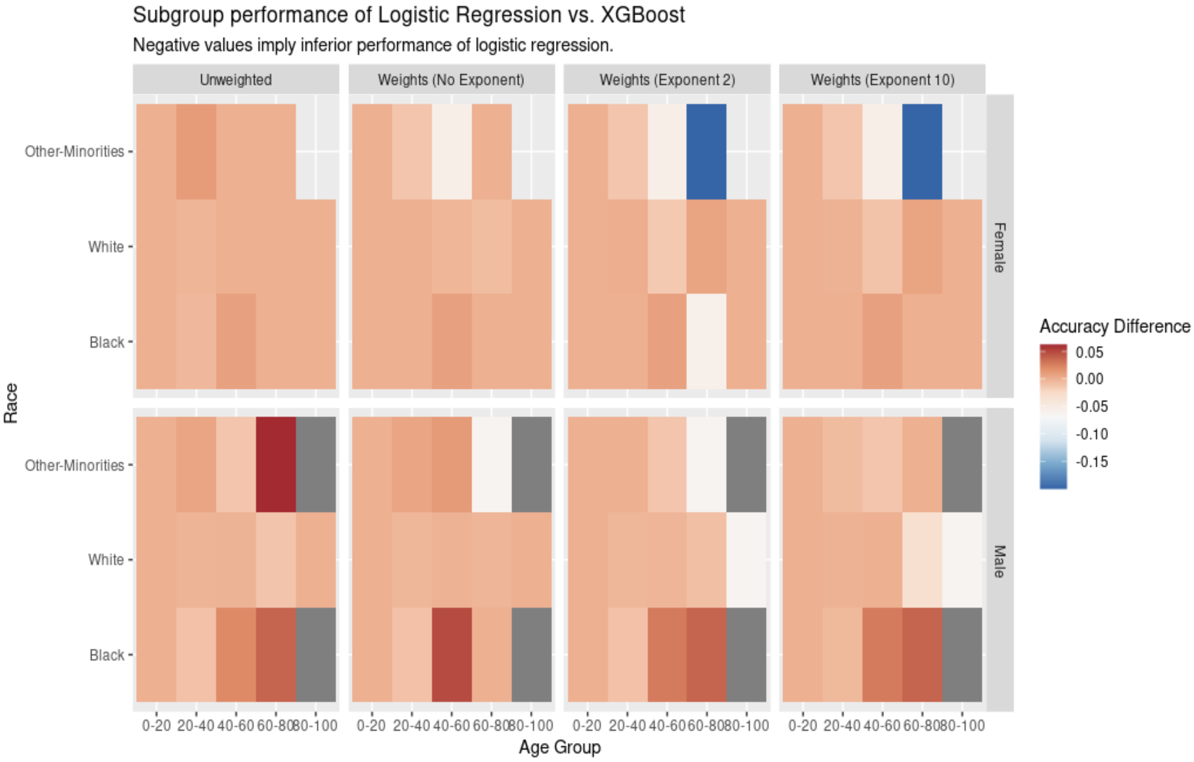

Figure 39: Subgroup Performance Difference between Logistic Regression and XGBoost (Acc)

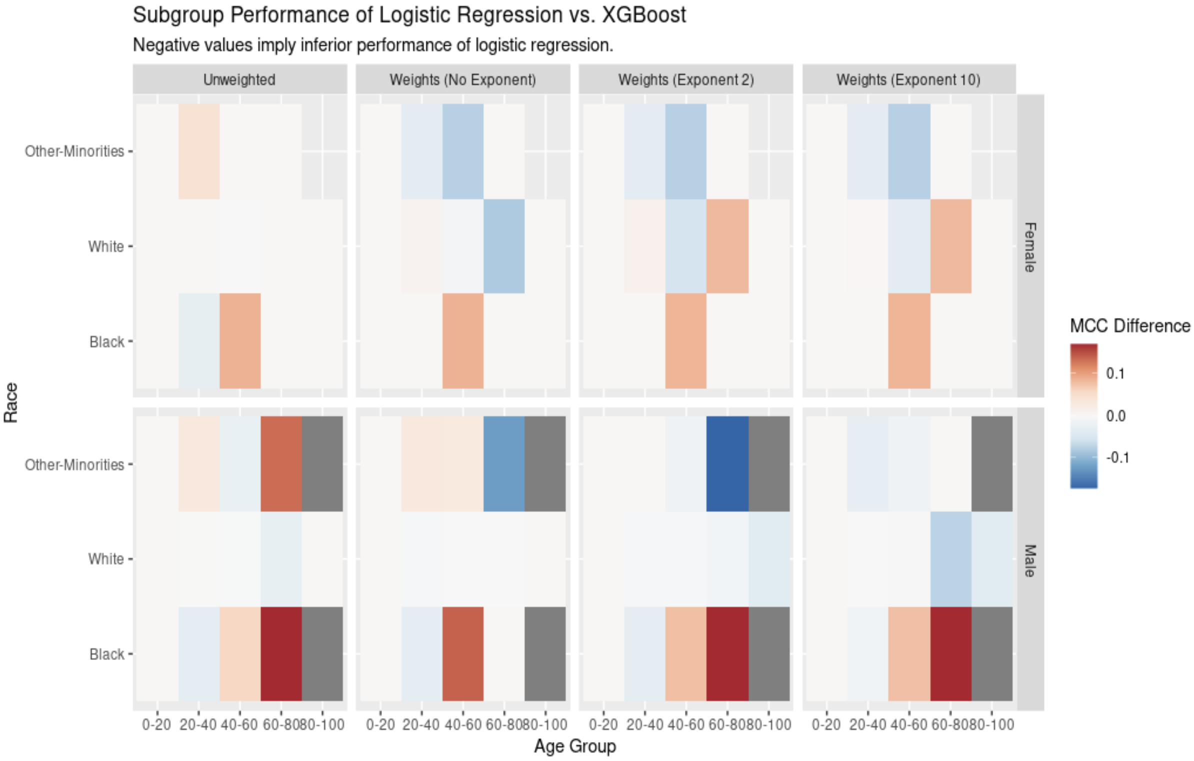

Figure 40: Subgroup Performance Difference between Logistic Regression and XGBoost (MCC)

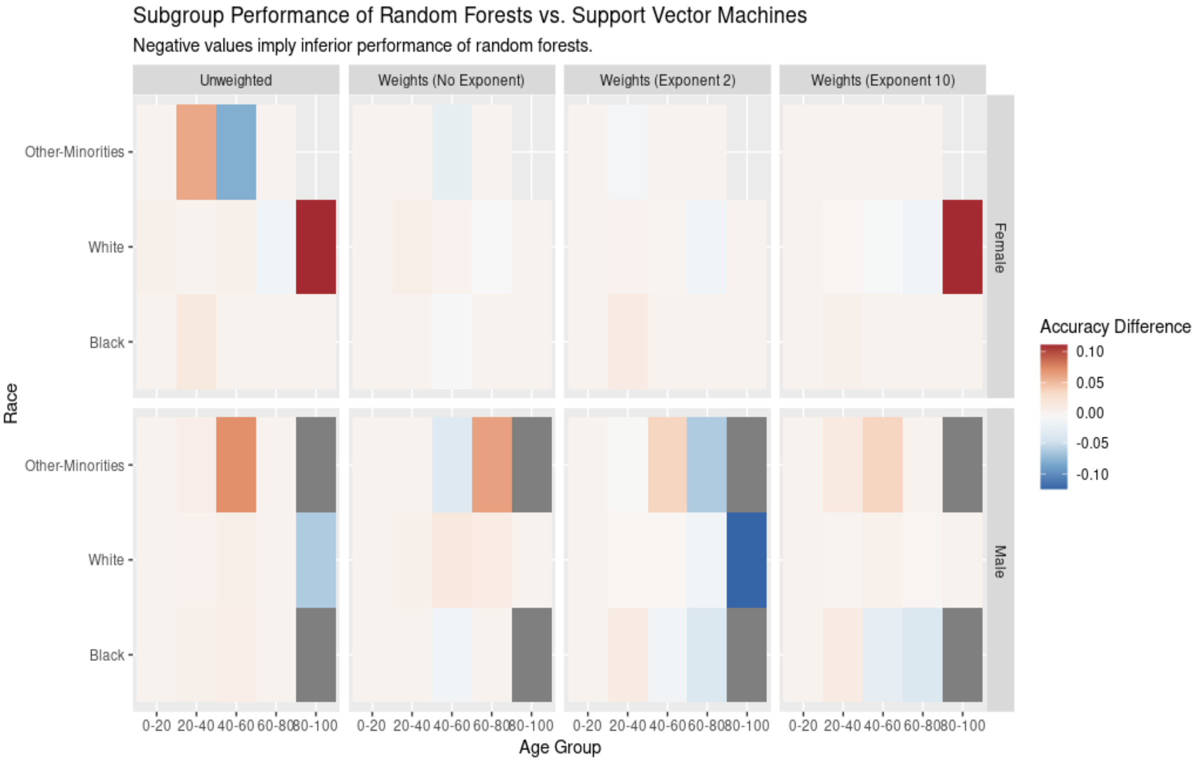

Figure 41: Subgroup Performance Difference between Random Forests and Support Vector Machines (Acc)

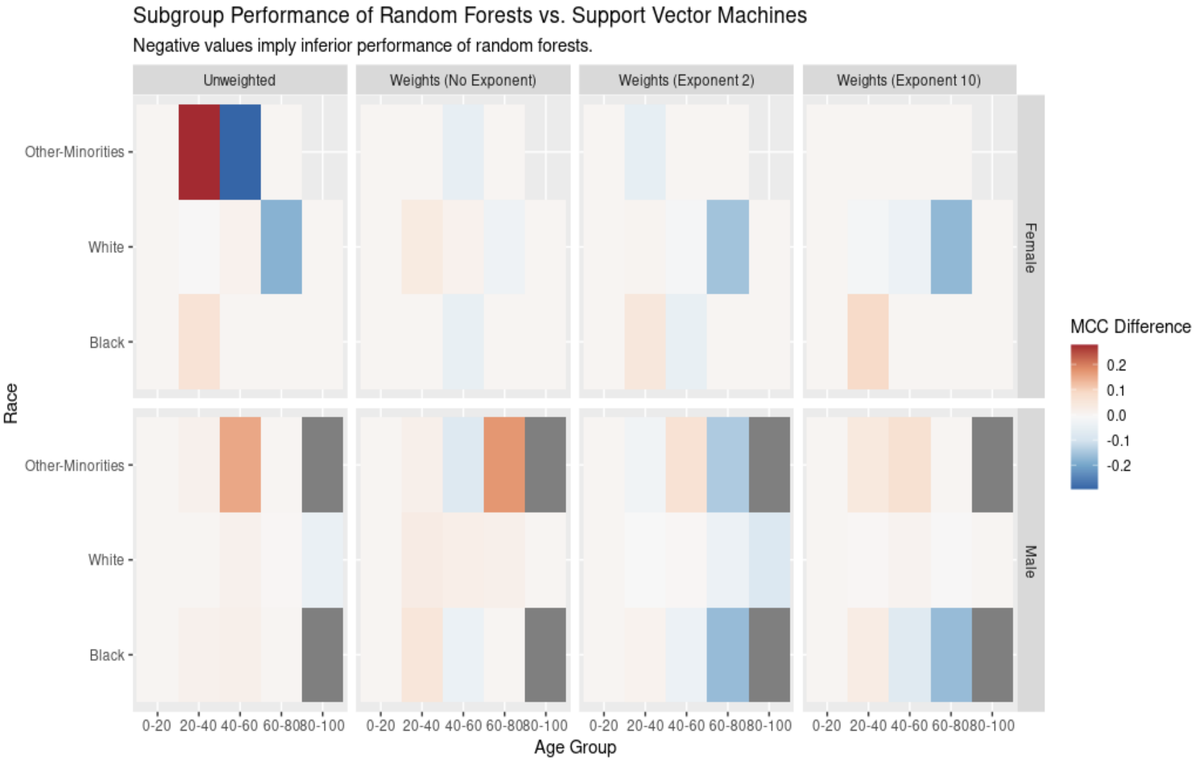

Figure 42: Subgroup Performance Difference between Random Forests and Support Vector Machines (MCC)

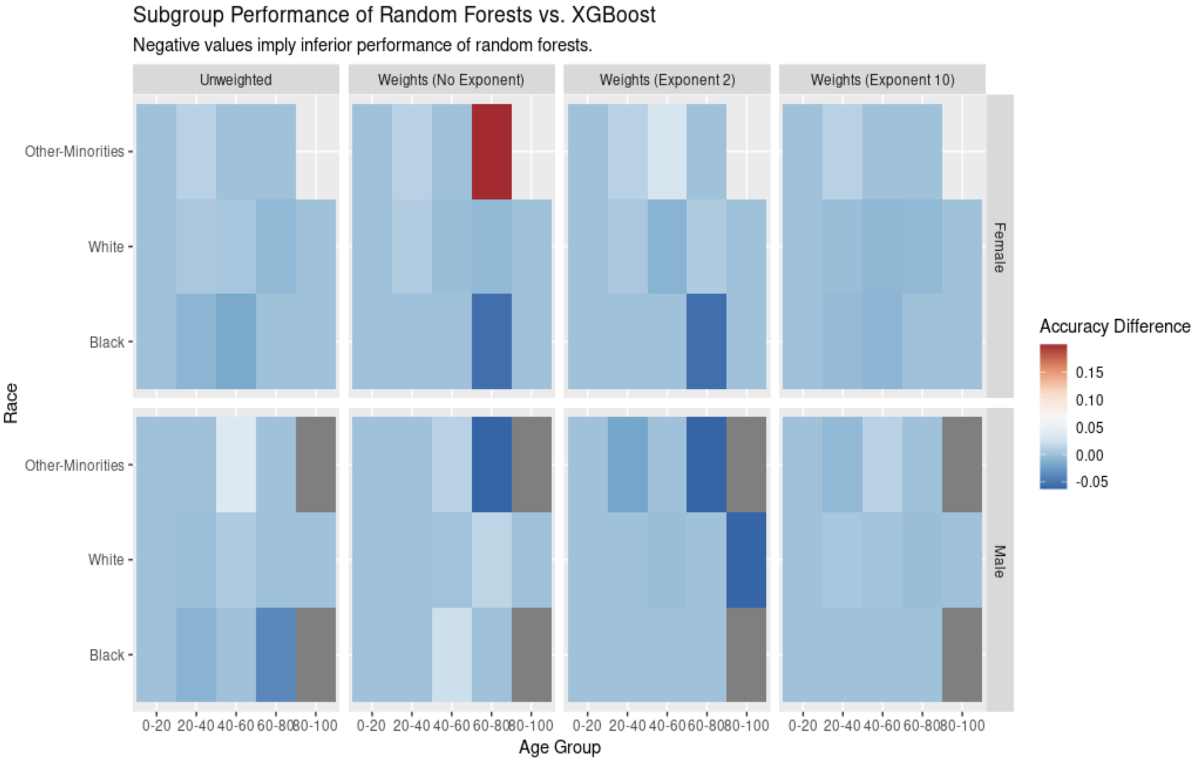

Figure 43: Subgroup Performance Difference between Random Forests and XGBoost (Acc)

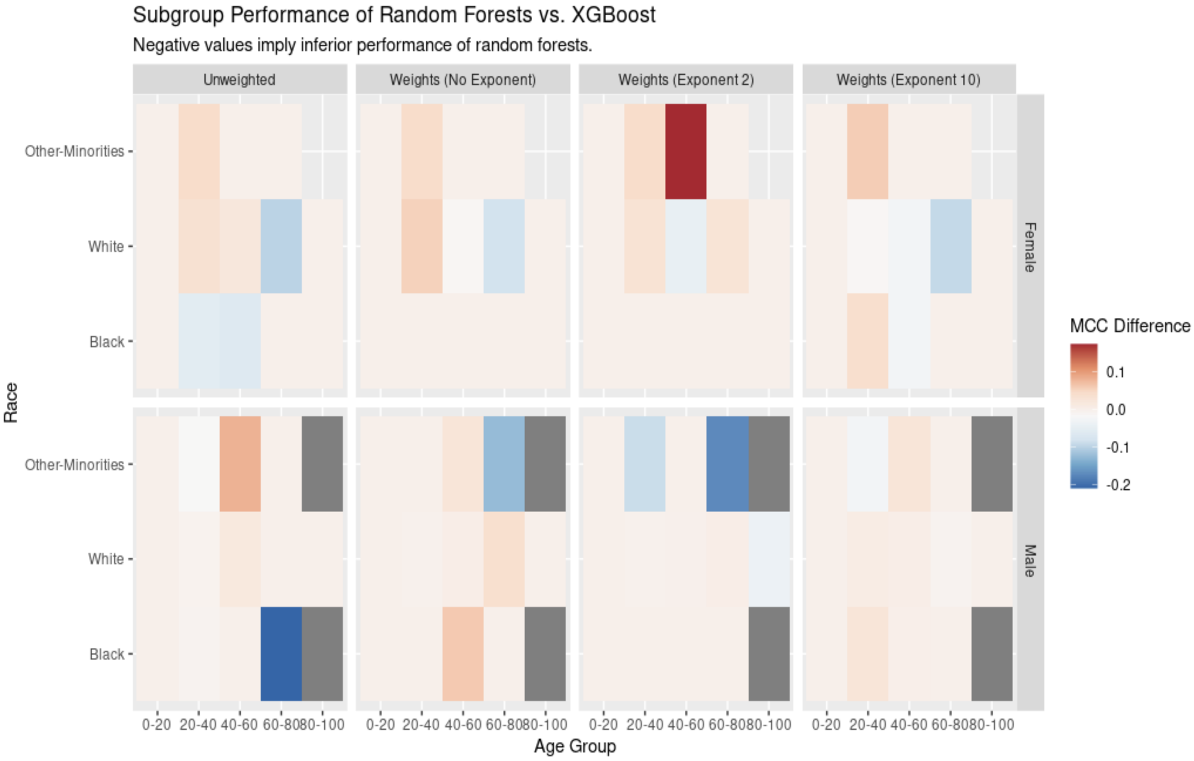

Figure 44: Subgroup Performance Difference between Random Forests and XGBoost (MCC)

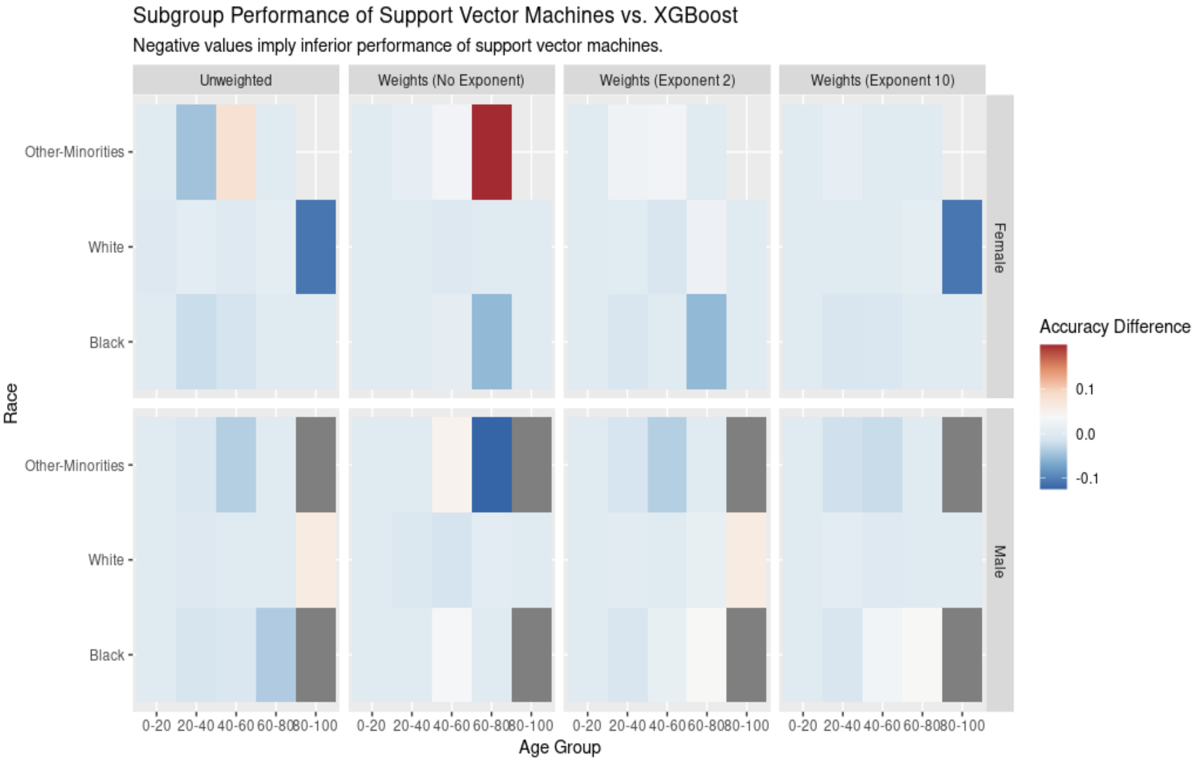

Figure 45: Subgroup Performance Difference between Support Vector Machines and XGBoost (Acc)

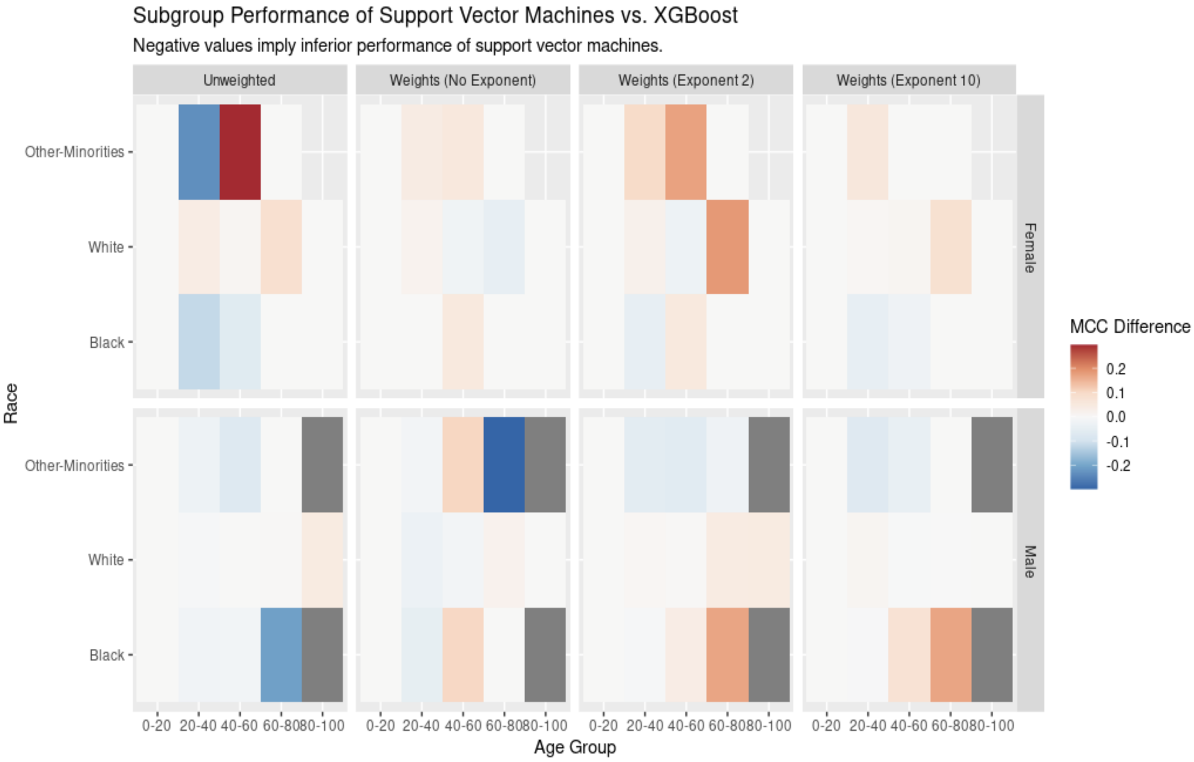

Figure 46: Subgroup Performance Difference between Support Vector Machines and XGBoost (MCC)

A.4.3 Overall Performance

Overall Accuracy				
Distribution Shift	*unweighted*	*exponent 1*	*exponent 2*	*exponent 3*
Logistic Regression	0.817	0.815	0.815	0.815
Random Forest	0.820	0.818	0.817	0.818
Support Vector Machines	0.816	0.813	0.820	0.818
XGBoost	0.818	0.817	0.818	0.818

Table 8: Overall Performance per Model and Shift (Acc)

Overall Matthew's Correlation Coefficient				
Distribution Shift	*unweighted*	*exponent 1*	*exponent 2*	*exponent 3*
Logistic Regression	0.465	0.460	0.462	0.463
Random Forest	0.471	0.466	0.464	0.468
Support Vector Machines	0.464	0.441	0.477	0.471
XGBoost	0.467	0.461	0.468	0.468

Table 9: Overall Performance per Model and Shift (MCC)

A.5 Subgroup Performance Results, Reduced Feature Set

A.5.1 Subgroup Performance per Classifier

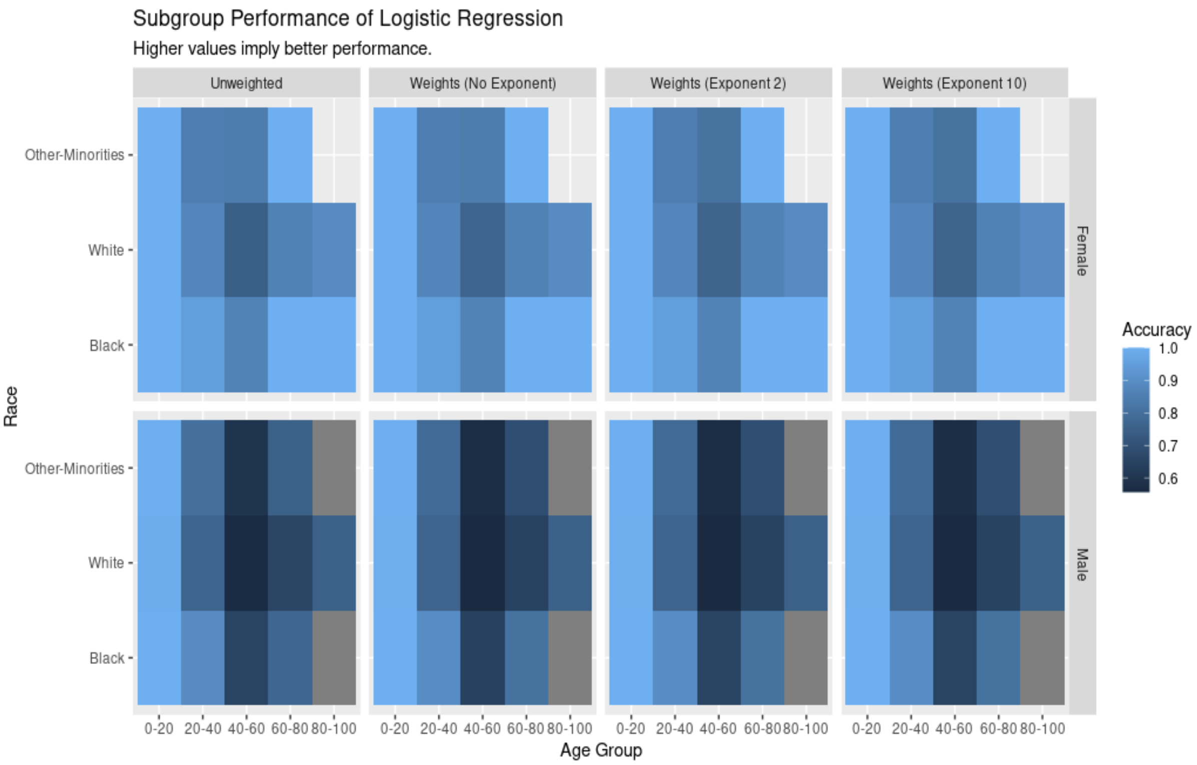

Figure 47: Subgroup Performance of Logistic Regression (Acc)

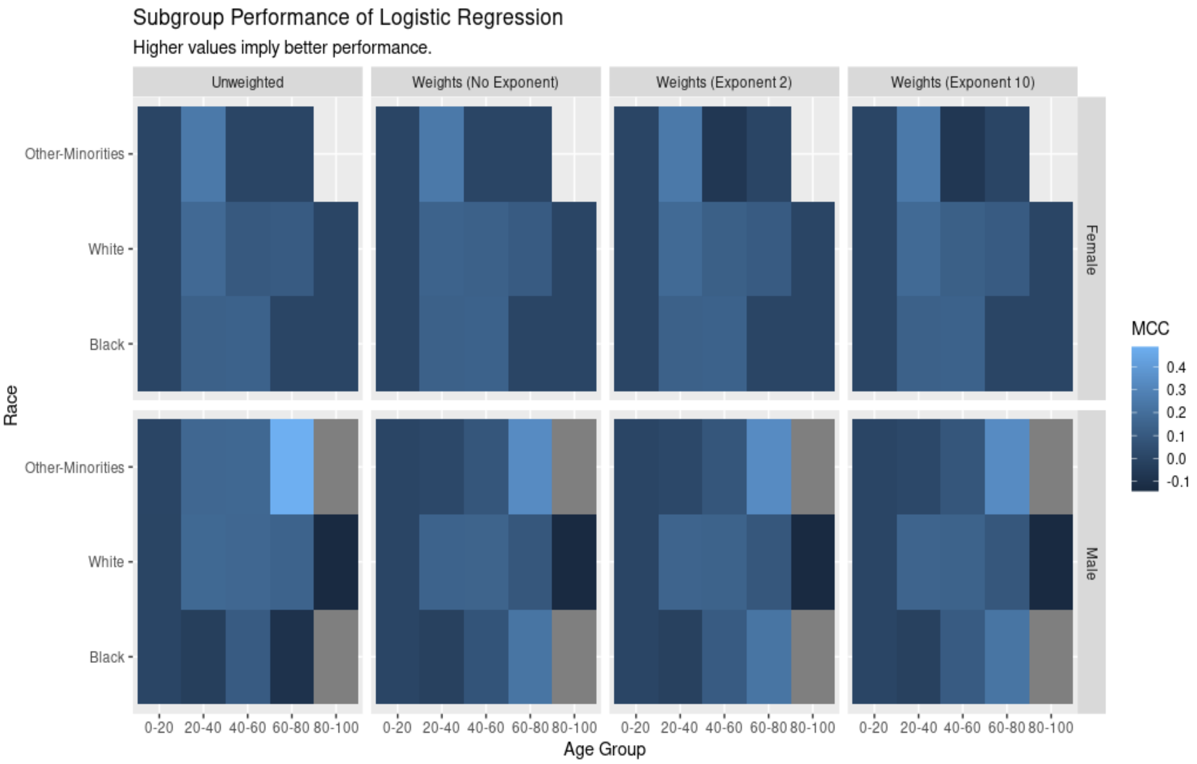

Figure 48: Subgroup Performance of Logistic Regression (MCC)

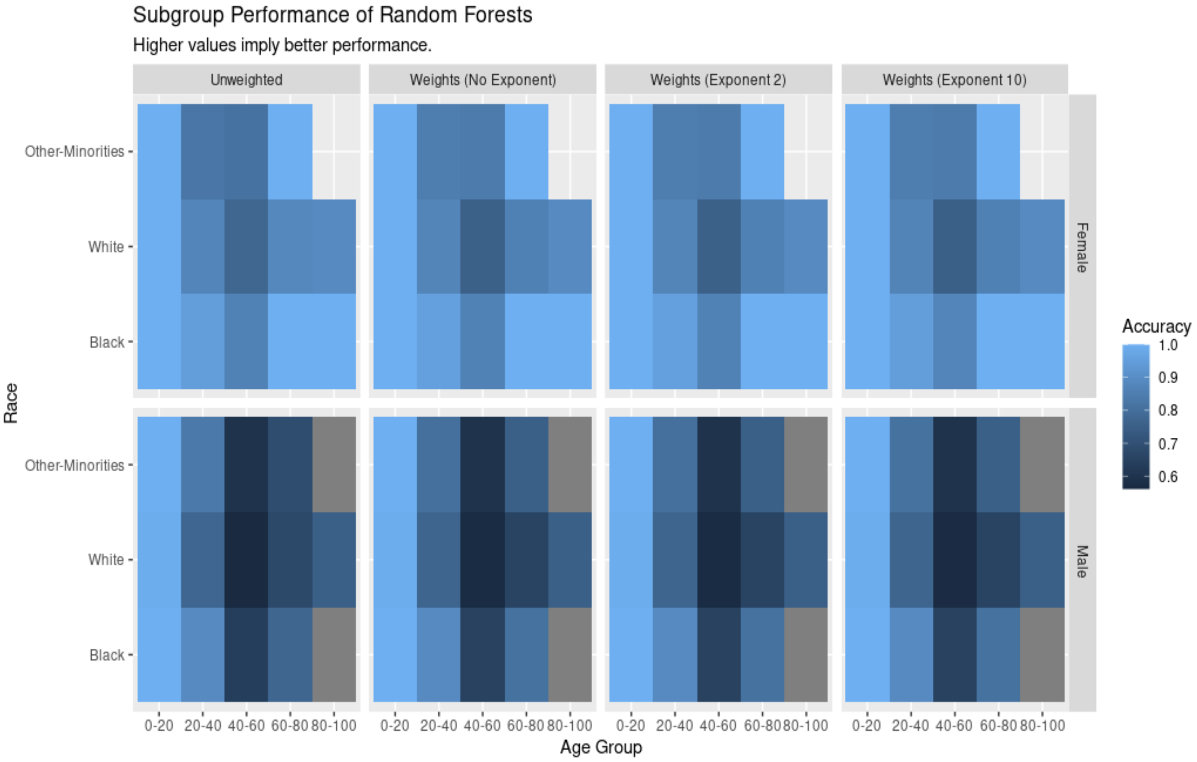

Figure 49: Subgroup Performance of Random Forests (Acc)

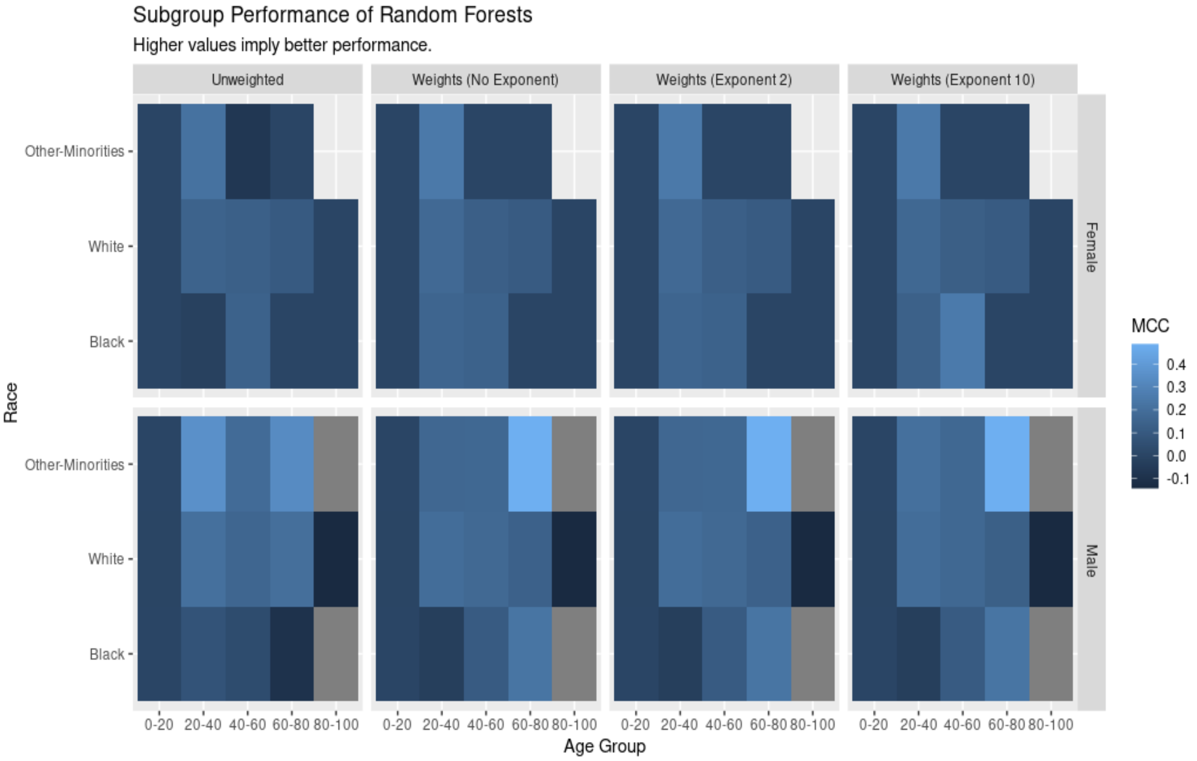

Figure 50: Subgroup Performance of Random Forests (MCC)

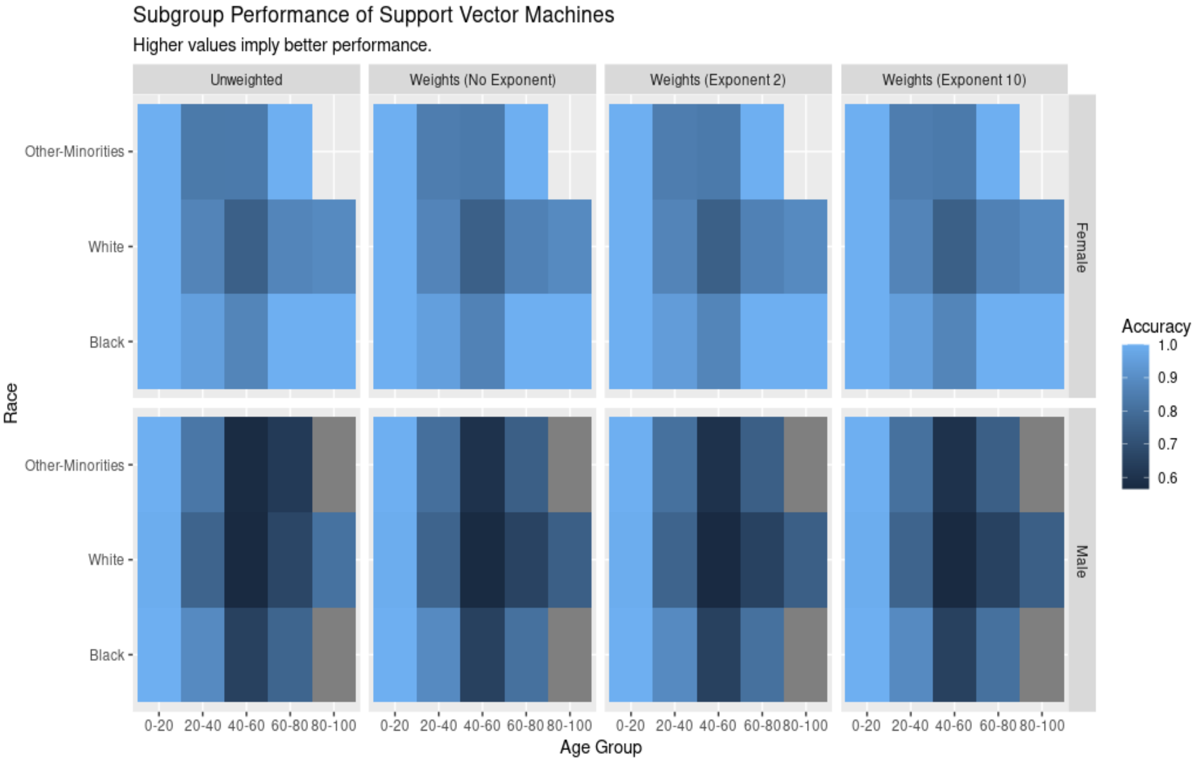

Figure 51: Subgroup Performance of Support Vector Machines (Acc)

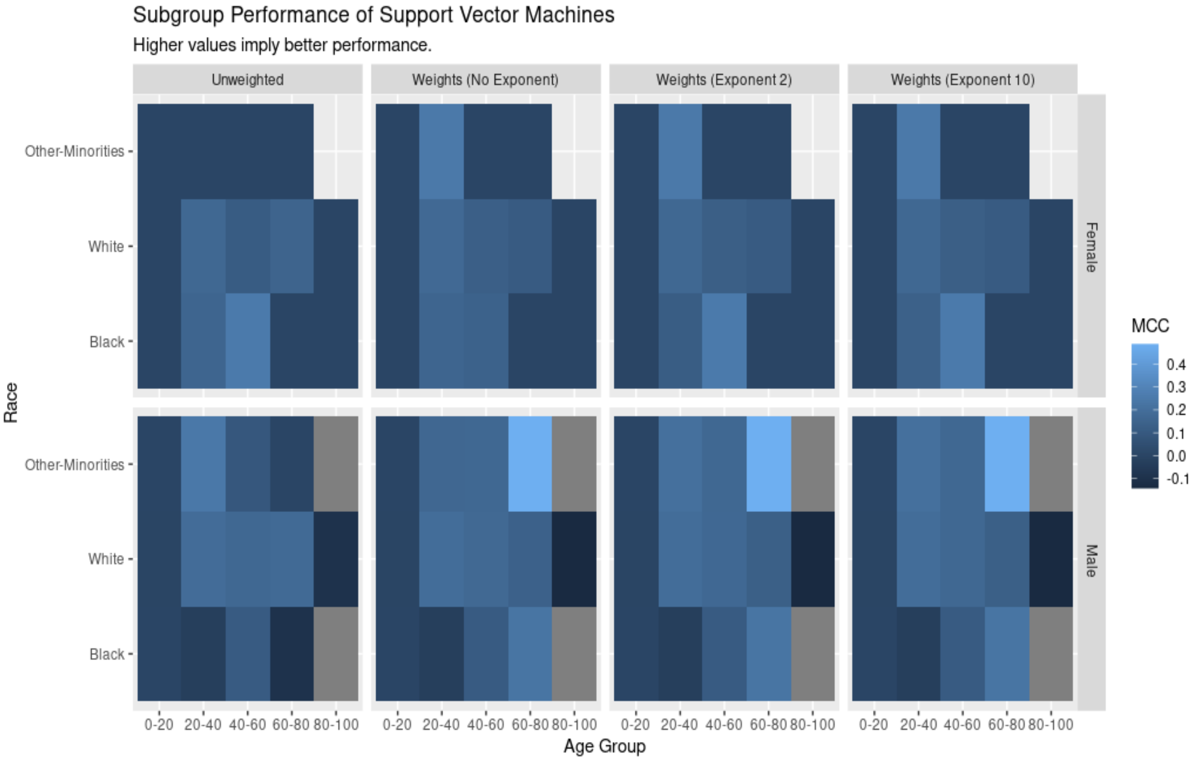

Figure 52: Subgroup Performance of Support Vector Machines (MCC)

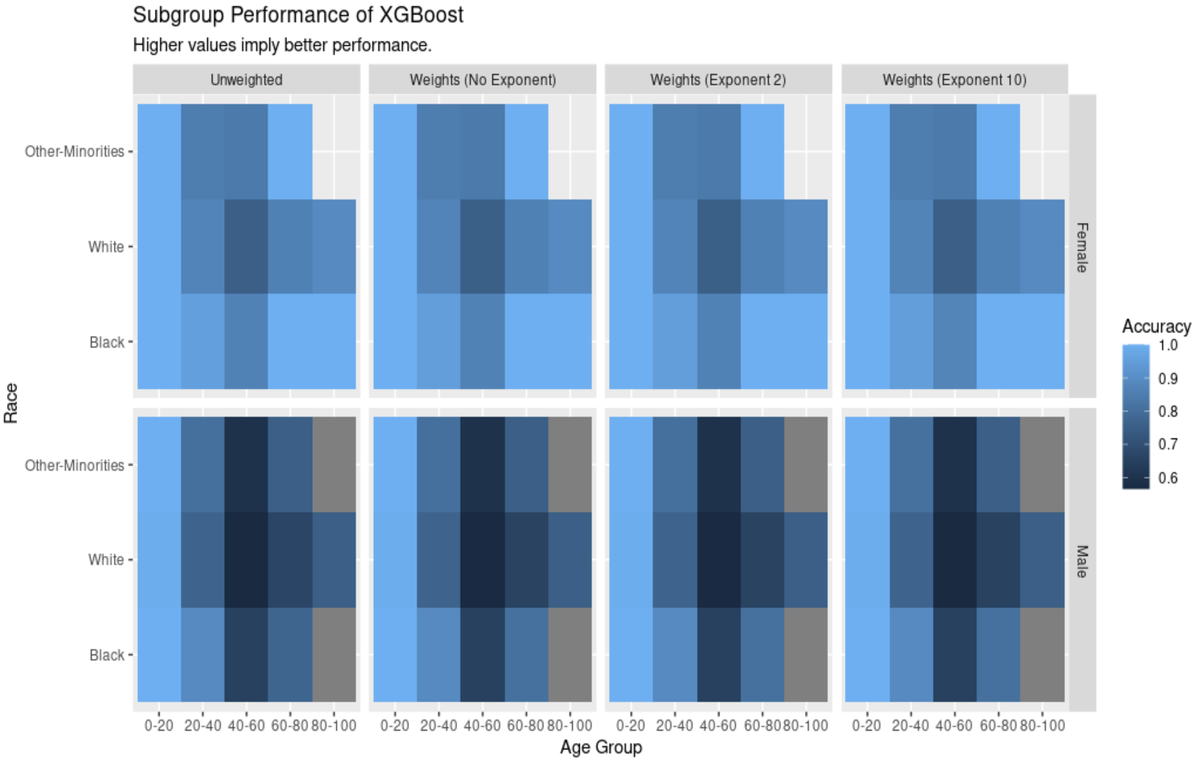

Figure 53: Subgroup Performance of XGBoost (Acc)

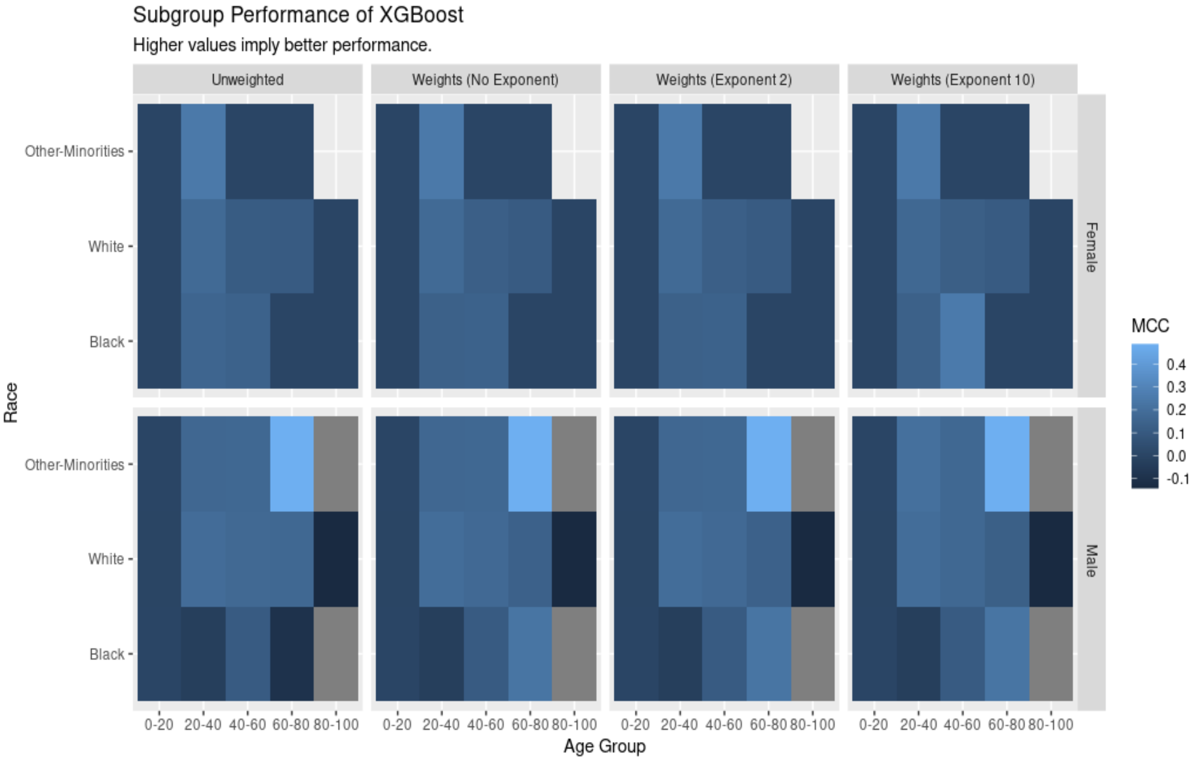

Figure 54: Subgroup Performance of XGBoost (MCC)

A.5.2 Classifier Comparison

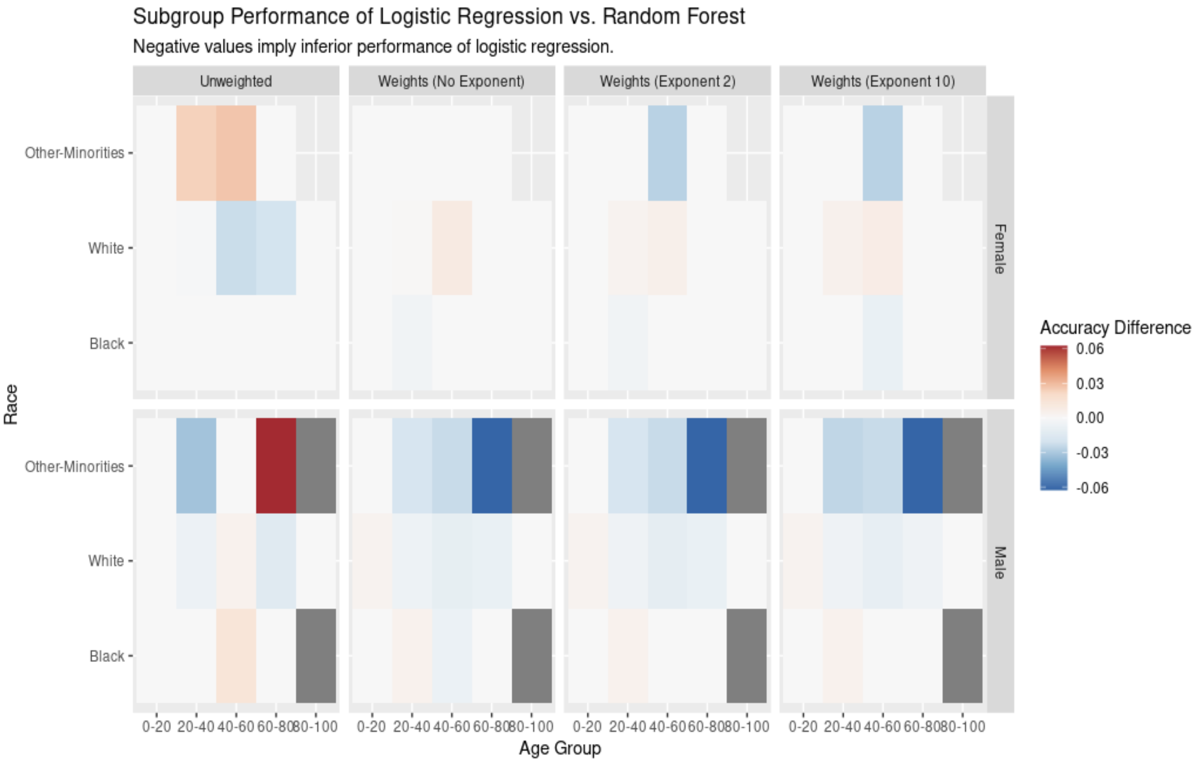

Figure 55: Subgroup Performance Difference between Logistic Regression and Random Forests (Acc)

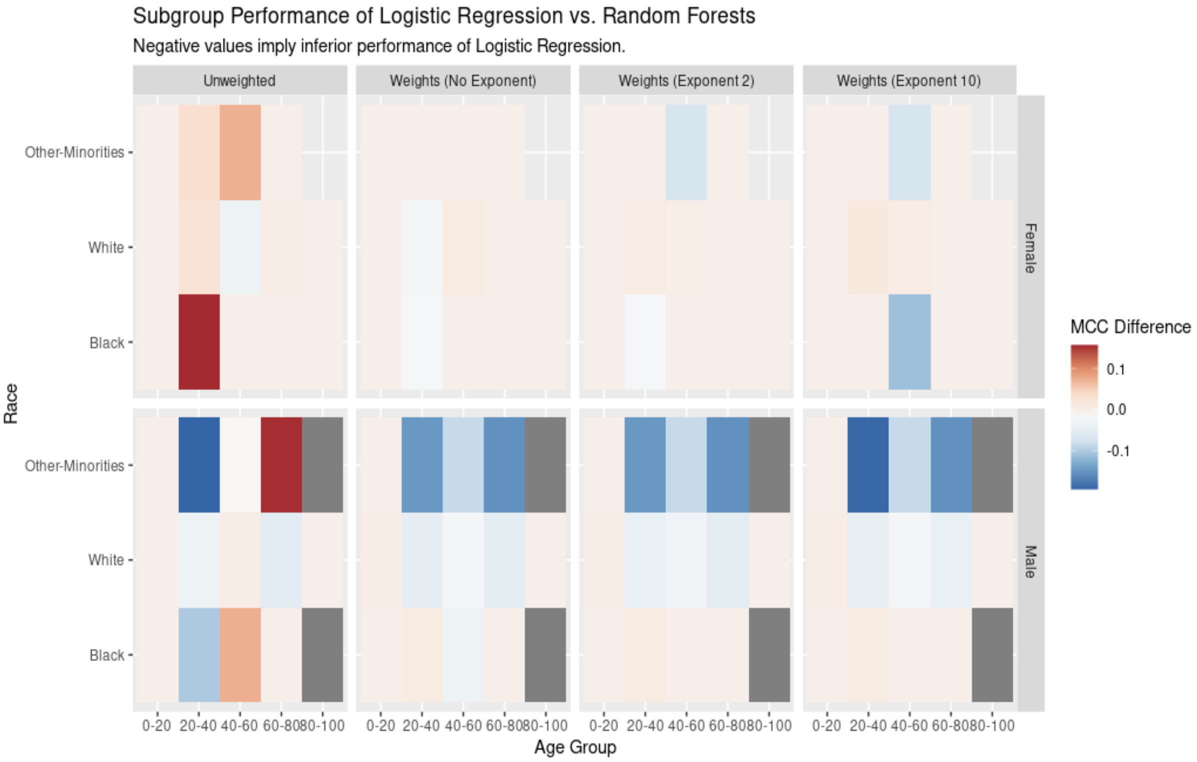

Figure 56: Subgroup Performance Difference between Logistic Regression and Random Forests (MCC)

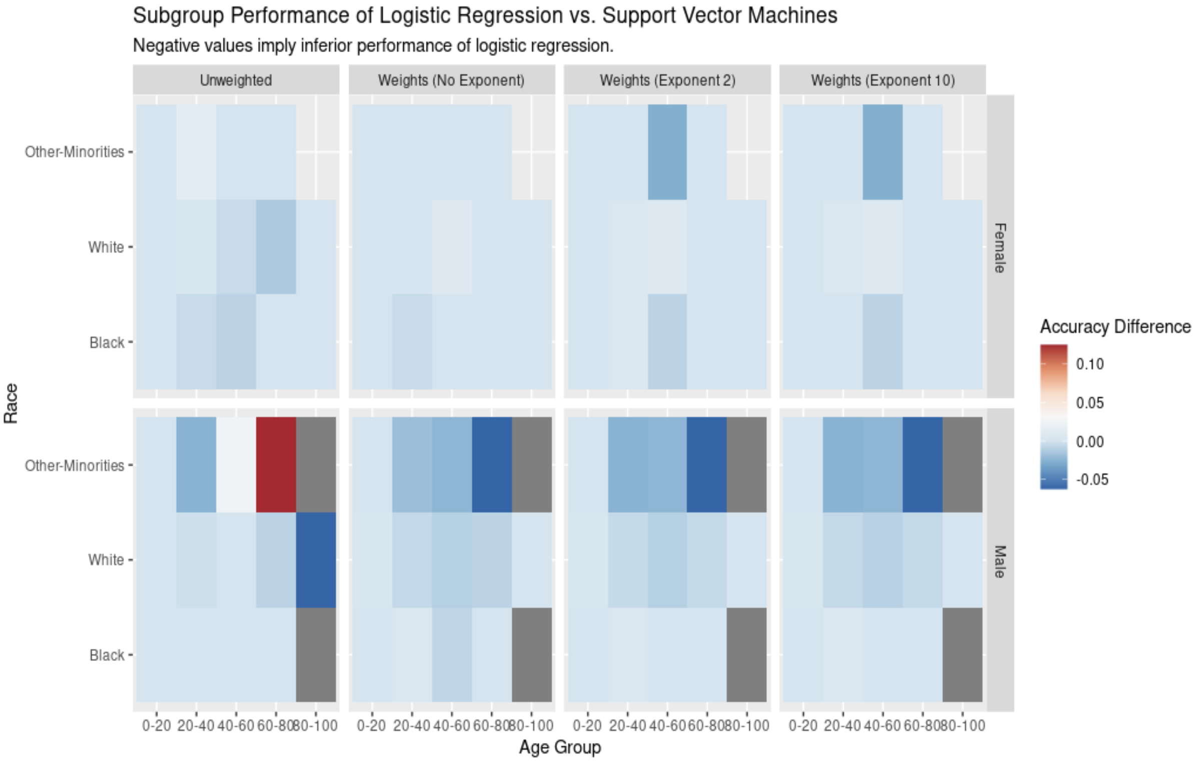

Figure 57: Subgroup Performance Difference between Logistic Regression and Support Vector Machines (Acc)

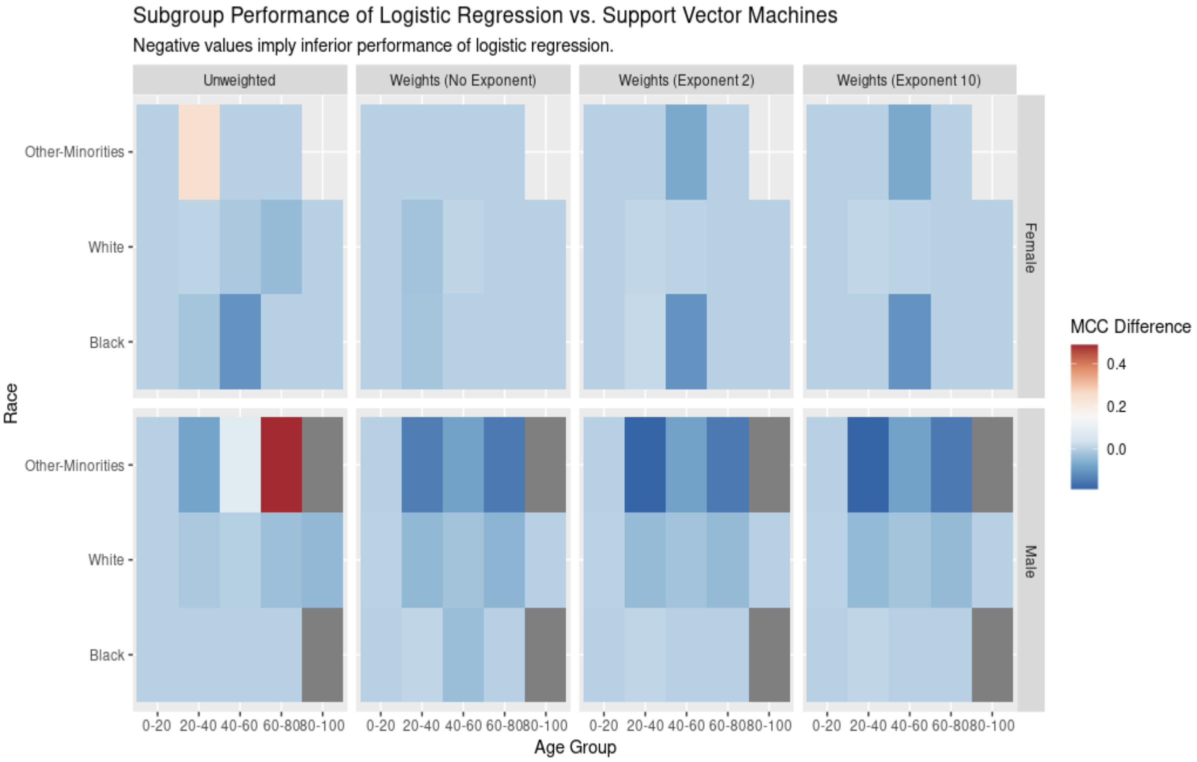

Figure 58: Subgroup Performance Difference between Logistic Regression and Support Vector Machines (MCC)

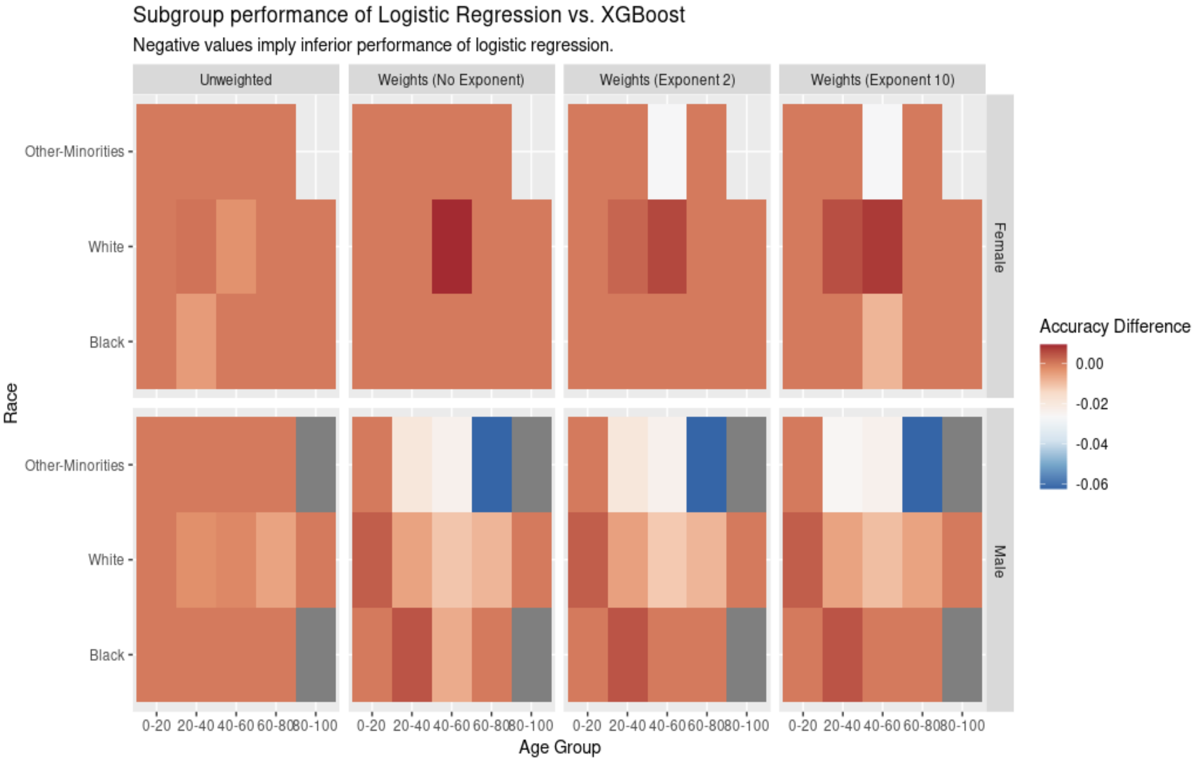

Figure 59: Subgroup Performance Difference between Logistic Regression and XGBoost (Acc)

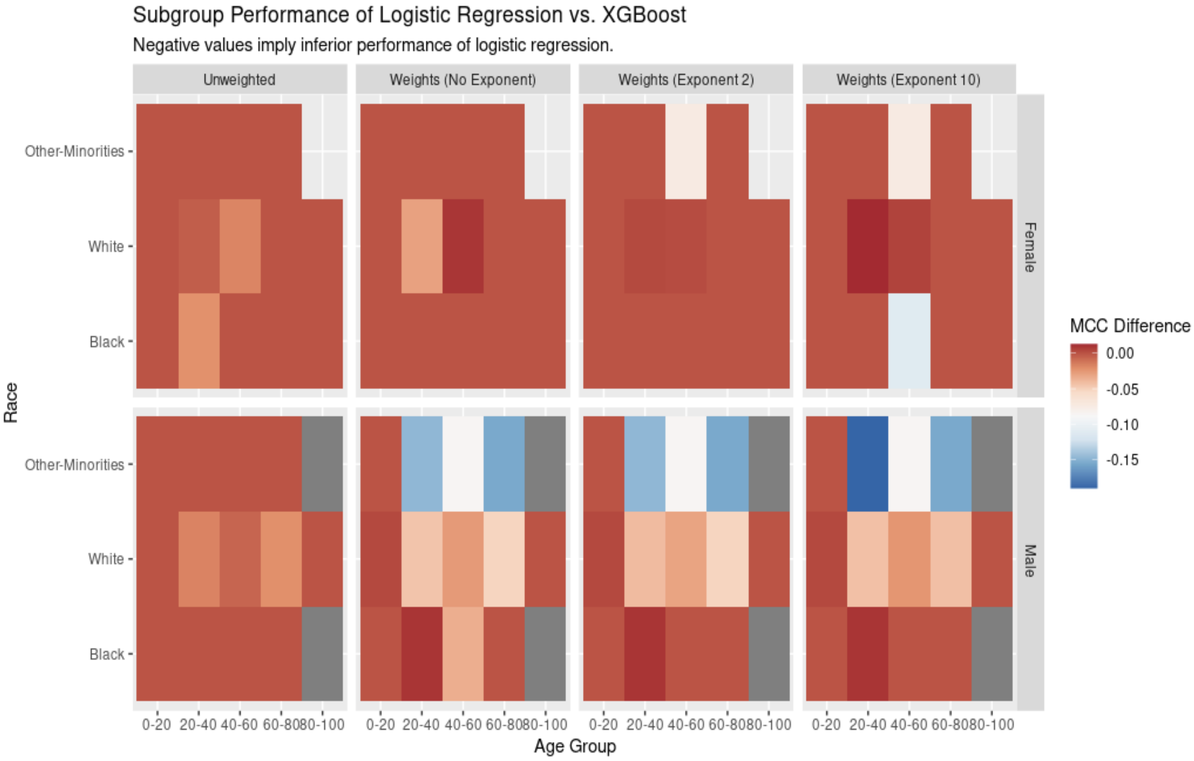

Figure 60: Subgroup Performance Difference between Logistic Regression and XGBoost (MCC)

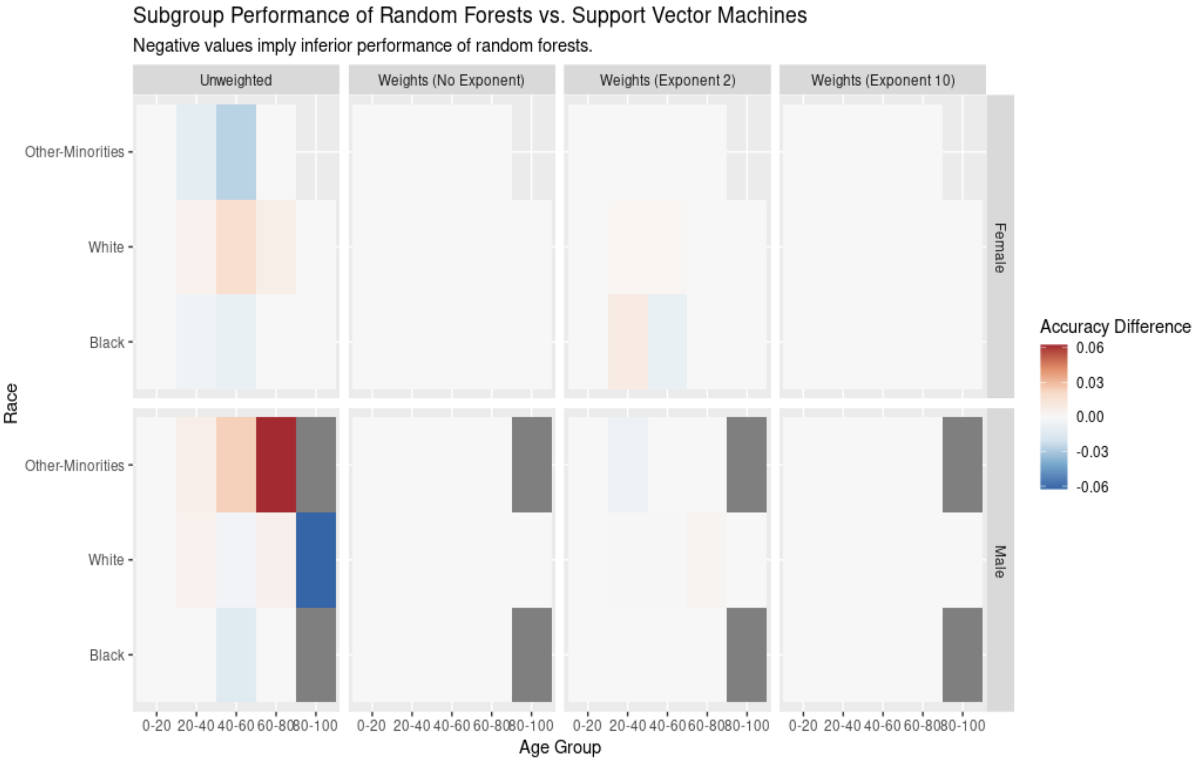

Figure 61: Subgroup Performance Difference between Random Forests and Support Vector Machines (Acc)

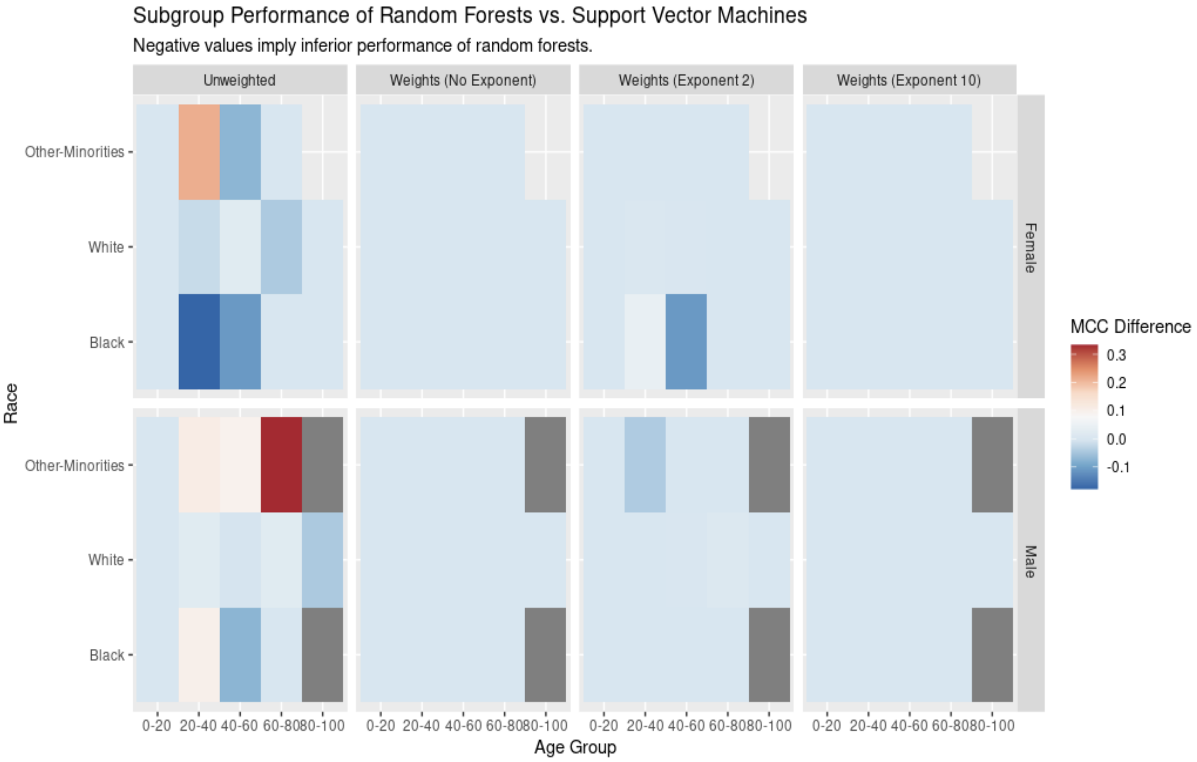

Figure 62: Subgroup Performance Difference between Random Forests and Support Vector Machines (MCC)

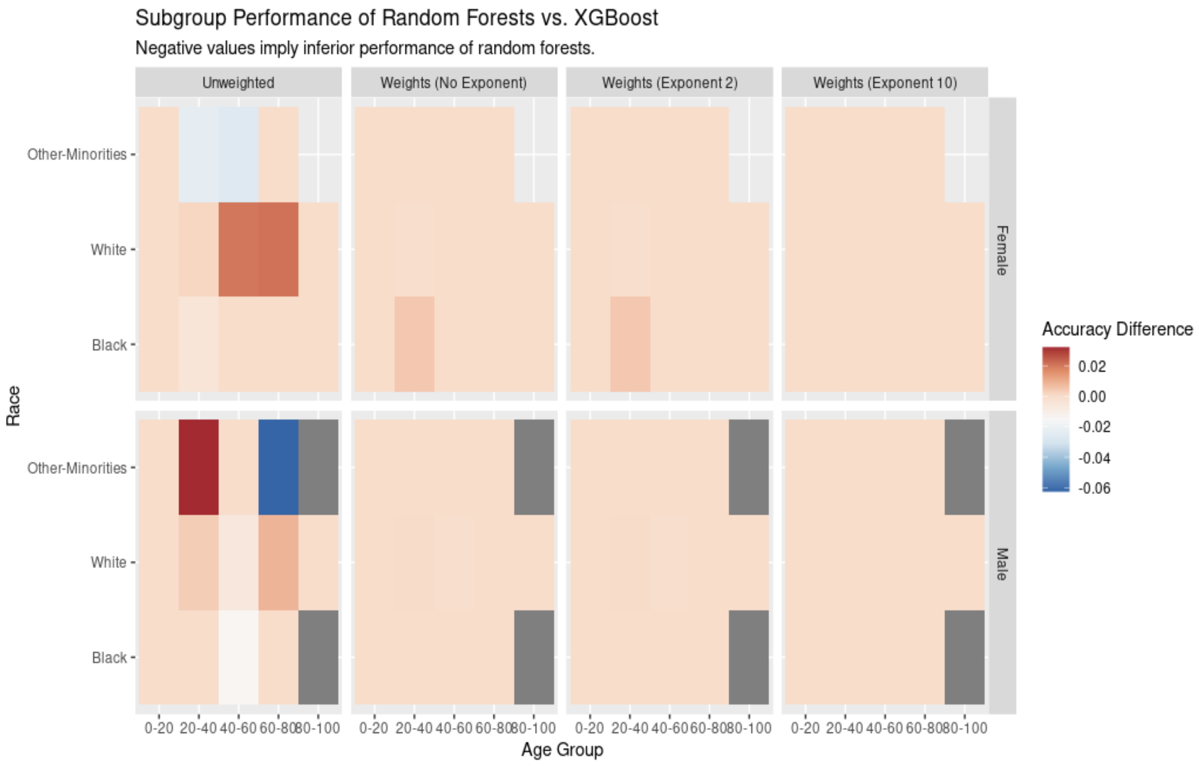

Figure 63: Subgroup Performance Difference between Random Forests and XGBoost (Acc)

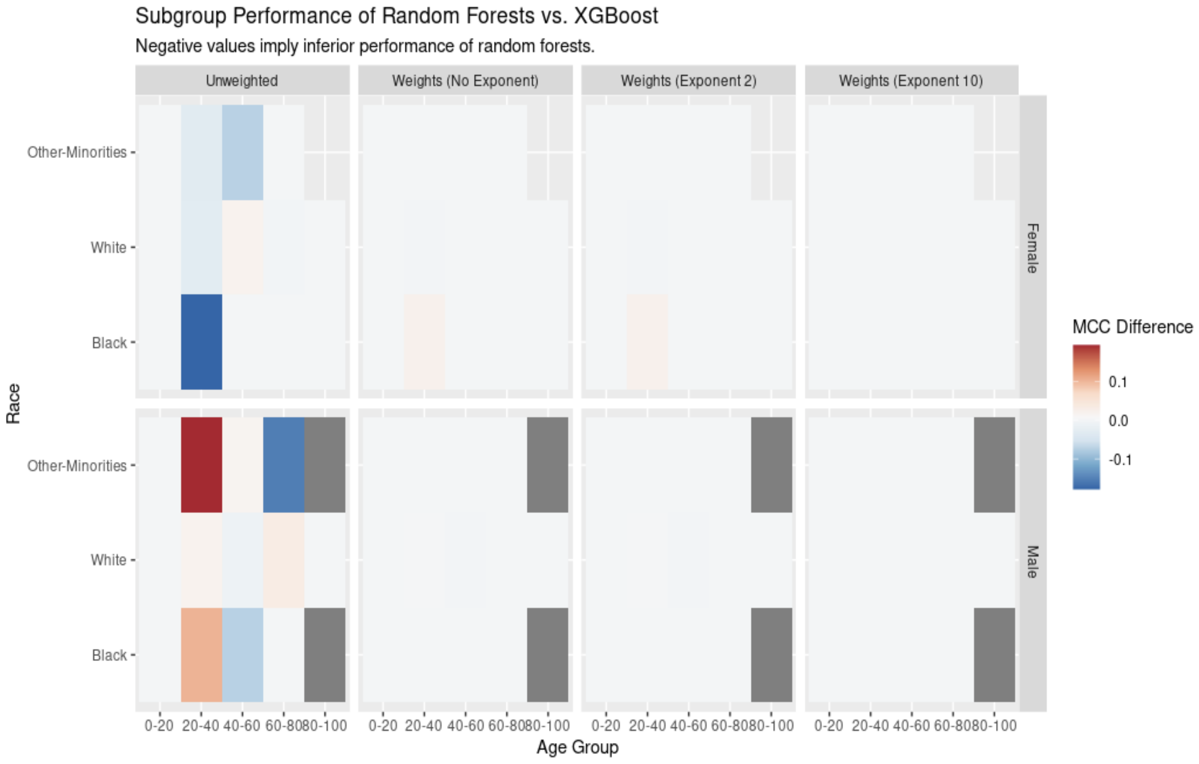

Figure 64: Subgroup Performance Difference between Random Forests and XGBoost (MCC)

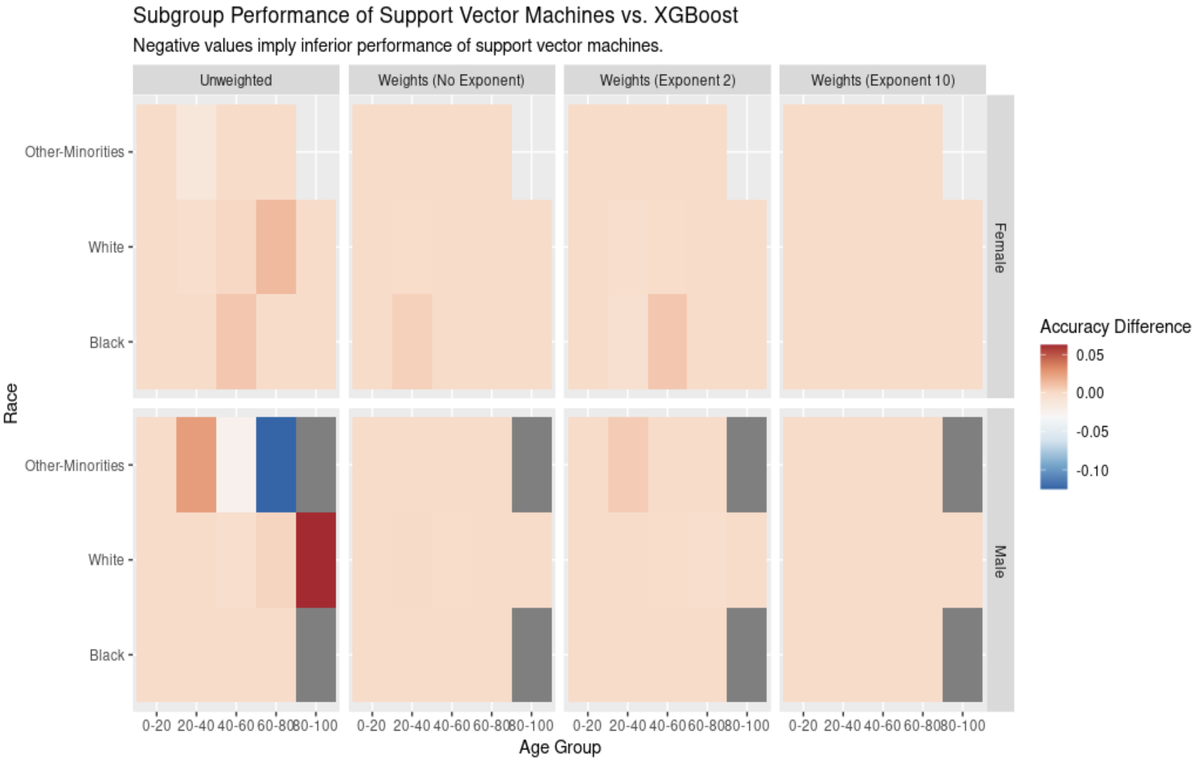

Figure 65: Subgroup Performance Difference between Support Vector Machines and XGBoost (Acc) (the graphics for all other classifiers can be found in the appendix)

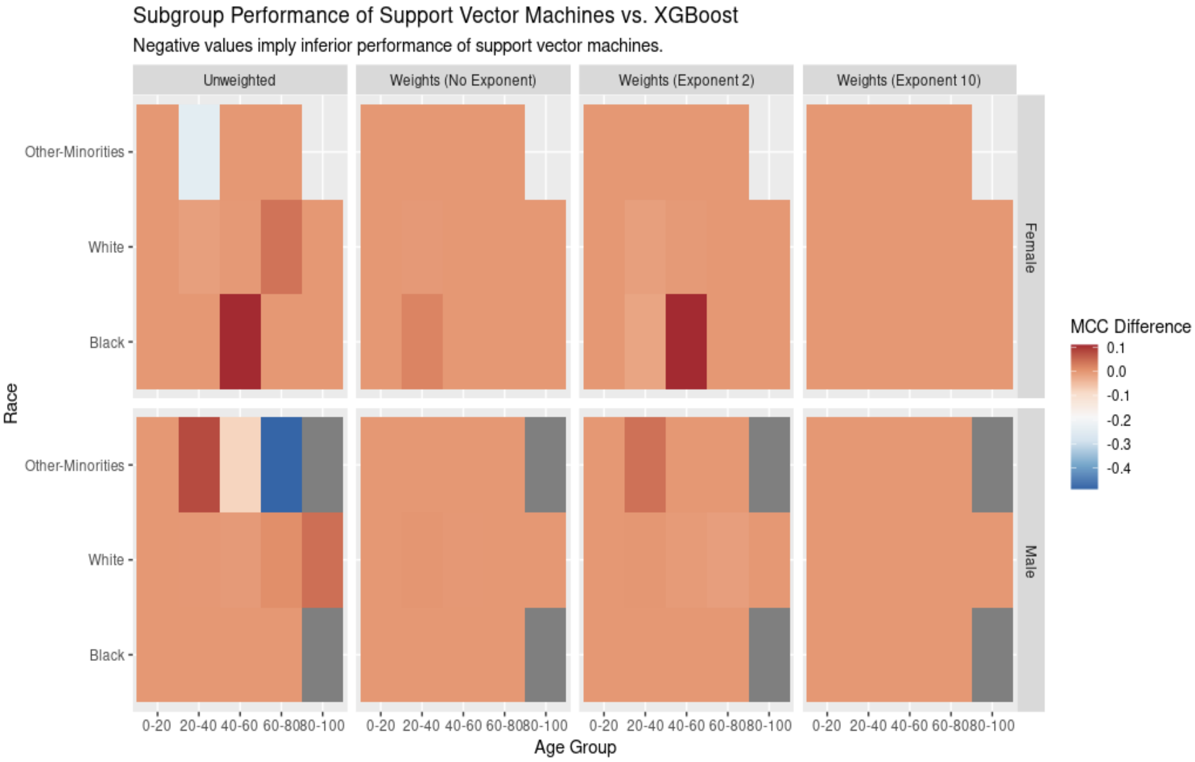

Figure 66: Subgroup Performance Difference between Support Vector Machines and XGBoost (MCC)

References

Antonelli, M., Johnston, E. W., Dikaios, N., Cheung, K. K., S Sidhu, H., Appayya, M. B., . . . Punwani, S. (2019). Machine learning classifiers can predict gleason pattern 4 prostate cancer with greater accuracy than experienced radiologists. *European Radiology*, *29*, 4754–4764. doi: 10.1007/s00330-019-06244-2

Balashankar, A., Lees, A., Welty, C., & Subramanian, L. (2019). *Pareto-efficient fairness for skewed subgroup data.* https://aiforsocialgood.github.io/icml2019/accepted/track1/pdfs/24_aisg_icml2019.pdf. (Accessed: 2022-11-27)

Batuwita, R., & Palade, V. (2013). Class imbalance learning methods for support vector machines. In *Imbalanced learning* (p. 83-99). John Wiley Sons, Ltd. doi: 10.1002/9781118646106.ch5

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(10), 281-305.

Binmore, K. G. (1981). *The foundations of topological analysis: A straightforward introduction.* Cambridge University Press. Retrieved from https://books.google.de/books?hl=de&lr=&id=o485AAAAIAAJ&oi=fnd&pg=PR10&dq=The+Foundations+of+Topological+Analysis:+A+Straightforward+Introduction&ots=kb2TALM4EM&sig=8fIb4GcNgHYHfOClce-rXq2H8Pg&redir_esc=y#v=onepage&q=The%20Foundations%20of%20Topological%20Analysis%3A%20A%20Straightforward%20Introduction&f=false

Bono, T., Croxson, K., & Giles, A. (2021). Algorithmic fairness in credit scoring. *Oxford Review of Economic Policy*, *37*, 585–617. doi: 10.1093/oxrep/grab020

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the 1st conference on fairness, accountability and transparency* (Vol. 81, p. 77-91). PMLR.

Caton, S., & Haas, C. (2020). *Fairness in machine learning: A survey.* arXiv. doi: 10.48550/ARXIV.2010.04053

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (p. 785–794). Association for Computing Machinery. doi: 10.1145/2939672.2939785

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., . . . Yuan, J. (2022). *Package 'xgboost'.* https://cran.r-project.org/web/packages/xgboost/xgboost.pdf. (Accessed: 2022-11-29)

Courvoisier, D. S., Combescure, C., Agoritsas, T., Gayet-Ageron, A., & Perneger, T. V. (2011). Performance of logistic regression modeling: beyond the number of events per variable, the role of data structure. *Journal of Clinical Epidemiology*, *64*(9), 993-1000. doi: https://doi.org/10.1016/j.jclinepi.2010.11.012

Creager, E., Madras, D., Jacobsen, J.-H., Weis, M., Swersky, K., Pitassi, T., & Zemel, R. (2019). Flexibly fair representation learning by disentanglement. In *Proceedings of the 36th international conference on machine learning* (Vol. 97, p. 1436-1445). PMLR.

Delgado, R., & Tibau, X.-A. (2019). Why cohen's kappa should be avoided as performance measure in classification. *PLOS ONE*, *14*, 1-26. doi: 10.1371/journal.pone.0222916

Denil, M., Matheson, D., & De Freitas, N. (2014). Narrowing the gap: Random forests in theory and in practice. In *Proceedings of the 31st international conference on machine learning* (Vol. 32, pp. 665–673). PMLR.

Ding, F., Hardt, M., Miller, J., & Schmidt, L. (2021). Retiring adult: New datasets for fair machine learning. In *Advances in neural information processing systems.*

Ernst, F., & Schweikard, A. (2020). *Fundamentals of machine learning.* UVK Verlag. doi: 10.36198/9783838552514

Fabris, A., Messina, S., Silvello, G., & Susto, G. A. (2022). Tackling documentation debt: A survey on algorithmic fairness datasets. In *Equity and access in algorithms, mechanisms, and optimization.* Association for Computing Machinery. doi: 10.1145/3551624.3555286

Fahrmeir, L., Kneib, T., & Lang, S. (2009). *Regression: Modelle, Methoden und Anwendungen.* Springer-Verlag GmbH. doi: 10.1007/978-3-642-01837-4

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. D. (2021). *Regression: Models, Methods and Applications.* Springer-Verlag GmbH. doi: 10.1007/978-3-662-63882-8

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, *15*(90), 3133–3181. Retrieved from http://jmlr.org/papers/v15/delgado14a.html

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets.* Springer. doi: 10.1007/978-3-319-98074-4

Ferri, C., Hérnandez-Orallo, J., & Modroiu, G. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*, 27 - 38. doi: 10.1016/j.patrec.2008.08.010

Fleisher, W. (2021). What's fair about individual fairness? In *AIES '21: Proceedings of the 2021 AAAI/ACM conference on AI, ethics, and society* (pp. 480–490). Association for Computing Machinery. doi: 10.1145/3461702.3462621

Foulds, J. R., Islam, R., Keya, K. N., & Pan, S. (2020). An intersectional definition of fairness. In *2020 IEEE 36th international conference on data engineering (ICDE)* (p. 1918-1921). doi: 10.1109/ICDE48307.2020.00203

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., & Roth, D. (2018). A comparative study of fairness-enhancing interventions in machine learning. In *FAT* '19: Proceedings of the conference on fairness, accountability, and transparency* (pp. 329–338). Association for Computing Machinery. doi: 10.1145/3287560.3287589

Gardner, J., Popović, Z., & Schmidt, L. (2022). *Subgroup robustness grows on trees: An empirical baseline investigation.* arXiv. doi: 10.48550/ARXIV.2211.12703

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). *Why do tree-based models still outperform deep learning on tabular data?* arXiv. doi: 10.48550/ARXIV.2207.08815

Hashimoto, T., Srivastava, M., Namkoong, H., & Liang, P. (2018). Fairness without demographics in repeated loss minimization. In *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 1929–1938). PMLR. Retrieved from http://proceedings.mlr.press/v80/hashimoto18a/hashimoto18a.pdf

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The elements of statistical learning: Data mining, inference, and prediction.* Springer. doi: 10.1007/b94608

Hebert-Johnson, U., Kim, M., Reingold, O., & Rothblum, G. (2018). Multicalibration: Calibration for the (Computationally-identifiable) masses. In *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 1939–1948). PMLR.

Hooker, S. (2021). Moving beyond "algorithmic bias is a data problem". *Patterns*, *2*(4), 100241. doi: https://doi.org/10.1016/j.patter.2021.100241

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). *A practical guide to support vector classification.* Taipei, Taiwan. Retrieved from https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Hutchinson, B., & Mitchell, M. (2018). 50 years of test (un)fairness: Lessons for machine learning. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 49–58). Association for Computing Machinery. doi: 10.1145/3287560.3287600

Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, *11*, 1–20. doi: 10.18637/jss.v011.i09

Kearns, M., Neel, S., Roth, A., & Wu, Z. S. (2018). Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 2564–2572). PMLR.

Kearns, M., Neel, S., Roth, A., & Wu, Z. S. (2019). An empirical study of rich subgroup fairness for machine learning. In *FAT* '19: Proceedings of the conference on fairness, accountability, and transparency* (pp. 100–109). Association for Computing Machinery. doi: 10.1145/3287560.3287592

Kim, M. P., Ghorbani, A., & Zou, J. (2018). Multiaccuracy: Black-box post-processing for fairness in classification. In *AIES '19: Proceedings of the 2019 AAAI/ACM conference on AI, ethics, and society* (pp. 247–254). Association for Computing Machinery. doi: 10.1145/3306618.3314287

Kim, M. P., Reingold, O., & Rothblum, G. N. (2018). Fairness through computationally-bounded awareness. In *NIPS'18: Proceedings of the 32nd international conference on neural information processing systems* (pp. 4847–4857). Curran Associates Inc. Retrieved from https://dl.acm.org/doi/10.5555/3327345.3327393

Kirasich, K., Smith, T., & Sadler, B. (2018). Random forest vs logistic regression: Binary classification for heterogeneous datasetsheterogeneous datase. *SMU Data Science Review*, *1*(3). Retrieved from https://scholar.smu.edu/datasciencereview/vol1/iss3/9

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., ... Liang, P. (2021, 18–24 Jul). Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th international conference on machine learning* (Vol. 139, pp. 5637–5664). PMLR.

Kohavi, R., & Becker, B. (1994). *Adult data set.* https://archive.ics.uci.edu/ml/datasets/adult. (Accessed: 2022-12-03)

Kuhn, H. W., & Tucker, A. W. (1951). Nonlinear programming. In *Proceedings of the second berkeley symposium on mathematical statistics and probability* (p. 481-492). Association for Computing Machinery. Retrieved from https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Second-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Nonlinear-Programming/bsmsp/1200500249

Kuhn, M. (2019). *The caret package.* Retrieved from https://topepo.github.io/caret/index.html

Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models.* Chapman and Hall/CRC. Retrieved from http://www.feat.engineering/

Kull, M., & Flach, P. A. (2014). *Patterns of dataset shift.* http://dmip.webs.upv.es/LMCE2014/Papers/lmce2014_submission_10.pdf. (Accessed: 2022-12-03)

Lahoti, P., Beutel, A., Chen, J., Lee, K., Prost, F., Thain, N., … Chi, E. (2020). Fairness without demographics through adversarially reweighted learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 728–740). Curran Associates, Inc.

Lum, K., Zhang, Y., & Bower, A. (2022). De-biasing "bias" measurement. In *2022 acm conference on fairness, accountability, and transparency* (p. 379–389). Association for Computing Machinery. doi: 10.1145/3531146.3533105

Maron, R. C., Schlager, J. G., Haggenmüller, S., von Kalle, C., Utikal, J. S., Meier, F., … Brinker, T. J. (2021). A benchmark for neural network robustness in skin cancer classification. *European Journal of Cancer*, *155*, 191–199. doi: https://doi.org/10.1016/j.ejca.2021.06.047

Martinez, N., Bertran, M., & Sapiro, G. (2020). Minimax pareto fairness: A multi objective perspective. In *Proceedings of the 37th international conference on machine learning* (Vol. 119, pp. 6755–6764). PMLR.

Martinez, N. L., Bertran, M. A., Papadaki, A., Rodrigues, M., & Sapiro, G. (2021). Blind pareto fairness and subgroup robustness. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (Vol. 139, pp. 7492–7501). PMLR.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, *54*, 1–35. doi: 10.1145/3457607

Morreno-Torres, J., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., & Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, *45*, 521–530. doi: 10.1016/j.patcog.2011.06.019

Murty, M., & Raghava, R. (2016). *Support vector machines and perceptrons: Learning, optimization, classification, and application to social networks.* Springer. doi: 10.1007/978-3-319-41063-0

Nguyen, J. (2017). *Logistic regression with UCI adult income.* Retrieved from https://www.kaggle.com/code/flyingwombat/logistic-regression-with-uci-adult-income/report (Accessed: 2022-11-29)

Nielsen, D. (2016). *Tree boosting with XGBoost - why does XGBoost win "every" machine learning competition?* Retrieved from https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2433761/16128_FULLTEXT.pdf doi: 10.48550/ARXIV.2207.08815

Owen, A. B. (2007). Infinitely imbalanced logistic regression. *Journal of Machine Learning Research*, *8*(27), 761–773.

Storkey, A. J. (2009). When training and test sets are different: Characterizing learning transfer. In J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, & N. D. Lawrence (Eds.), *Dataset shift in machine learning* (p. 3-28). MIT Press.

Sweeney, C., Ennis, E., Bond, R., Mulvenna, M. D., & O'Neill, S. (2021). Understanding a happiness dataset: How the machine learning classification accuracy changes with different demographic groups. In *2021 IEEE symposium on computers and communications (ISCC)* (p. 1-4). doi: 10.1109/ISCC53001.2021.9631455

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., & Schmidt, L. (2020). Measuring robustness to natural distribution shifts in image classification. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 18583–18599). Curran Associates, Inc.

Wainberg, M., Alipanahi, B., & Frey, B. J. (2016). Are random forests truly the best classifiers? *Journal of Machine Learning Research*, *17*(110), 1–5.

Wang, H., Hong, J., Zhou, J., & Wang, Z. (2022). *How robust is your fairness? evaluating and sustaining fairness under unseen distribution shifts.* arXiv. doi: 10.48550/ARXIV.2207.01168

Weisstein, E. W. (n.d.). *Hyperplane.* https://mathworld.wolfram.com/Hyperplane.html. (Accessed: 2022-11-29)

Wyner, A. J., Olson, M., Bleich, J., & Mease, D. (2017). Explaining the success of adaboost and random forests as interpolating classifiers. *Journal of Machine Learning Research*, *18*(48), 1–33.

XGBoost documentation. (2022). *Introduction to boosted trees.* https://xgboost.readthedocs.io/en/latest/tutorials/model.html. (Accessed: 2022-12-02)

Xiao, Y., & Watson, M. (2019). Guidance on conducting a systematic literature review. *Journal of Planning Education and Research*, *39*, 93–112. doi: 10.18637/jss.v011.i09

Zheng, W., & Jin, M. (2020). The effects of class imbalance and training data size on classifier learning: An empirical study. *SN COMPUT. SCI.*, *1*(71). doi: 10.1007/s42979-020-0074-0

Declaration of Authenticity

The work contained in this thesis is original and has not been previously submitted for examination which has led to the award of a degree.
To the best of my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made.

Theresa Kriecherbauer